# Calculation of $6j$ symbols

A thesis
submitted for the Degree
of
Doctor of Philosophy in Physics
in the
University of Canterbury

by

B.G. Searle

University of Canterbury
December 1988

# Abstract

At present there is no complete algorithm for the calculation of the coupling or recoupling factors of an arbitrary compact group. However, this thesis is based on the premise that the $6j$ can be calculated using only the Kronecker product rules for the group and the general relations between $6j$. We review the symmetry properties of the $6j$ symbols and choose a set of values for the permutation matrices of a mixed symmetry triad.

An algorithm is presented for the recursive calculation of coupling and recoupling factors in terms of the primitive factors. This algorithm is shown to be complete. It is then shown how this algorithm may be applied to a larger class of group theoretic transformation factors.

The primitive $6j$ are then split into four classes. This allows us to specify complete algorithms for the calculation of all but one of these classes. This is a major advance since it was previously necessary to systematically try all equations in order to solve an unknown $6j$.

We conjecture that our algorithm for the calculation of the fourth class of primitive $6j$, the core $6j$, is complete, although we are only able to prove this for $SO_3$. As a consequence we discuss the various special cases that occur in groups more complex than $SO_3$, starting with the point groups. New results are given for the groups $G_2$ and $E_8$, and the $6j$ for the mixed symmetry finite group $K_{20}$ are completely solved.

The data structures necessary for the implementation of the algorithms in a PASCAL program are discussed, along with the algorithms required to calculate the symmetrised powers of an irrep.

# Acknowledgement

I wish to acknowledge the financial support of a Roper Scholarship which enabled me to undertake this research.

I would like to express my gratitude to my supervisor Dr. P.H. Butler for valuable discussions and guidance.

I wish to than k my fellow research students, especially H.J. Ross and M.D. Albrow for some valuable discussions and ideas.

I would also like to acknowledge the support of my family.

# Contents

# List of Tables

# Chapter 1

# Introduction

The usefulness of group theory in physics has become well established. Groups provide a compact way of describing the symmetries of the various states of a system and are extensively used in quantum mechanics. The coupling and recoupling factors for a group and a chain of subgroups of that group then provide a convenient way of simplifying calculations involving couplings of the wavefunctions of the system. The symmetrised symbols, the $6j$ and $3jm$, are usually calculated instead of the coupling and recoupling factors due to their more convenient properties, although different, sometimes equivalent, definitions exist for these symbols.

At present no general algorithm exists for the complete calculation of the $6j$ or $3jm$. Most recent work has been concerned with the calculation of relatively simple multiplicity free coupling factors for various specific groups. Recent calculations of $6j$ includes Haase and Dirl(1986) for the symmetric groups, Haase and Butler(1985), Judd(1986,1987), Judd et al (1986) and Pluhar et al (1986) for the classical Lie groups, and Zeng(1987) for OSp(1,2). Coupling factors have been calculated by Chen et al(1985) for the space groups and have been used by Raynal and Conte(1985) to label the point groups.

This study originated in a request by C.Hamer (see Hamer, 1986) for us to extend the table of $SU_3$ $6j$. In attempting to achieve this it became clear that the program RACAH used by Butler(1981) to produce tables of $6j$ and $3jm$ for the point groups could be significantly improved. The calculation of primitive $6j$ was particularly inefficient since the program searched all possible equations involving the $6j$ until a result was found or the search failed. It was therefore decided to attempt to extend the algorithm proposed by Butler and Wybourne(1976), in an attempt to produce a complete algorithm for the calculation of $6j$ for any compact group. At the same time the program RACAH was rewritten in PASCAL.

Most other methods calculate $3jm$ from explicitly symmetrised basis functions and then calculate the $6j$ from the $3jm$. This requires a basis choice for the partner of each irrep to calculate the $3jm$, even though the $6j$ are totally independent of this basis. What has become known as Butler's method takes a more direct approach to the calculation of $6j$, using the Biedenharn-Elliott, Racah backcoupling and orthonormality relations for $6j$, which are valid for all compact groups (see Derome and Sharp 1965, Butler 1975). As well as these relations we require particular information about the group, specifically the product rules and symmetrised powers of the irreps. With this present approach the calculation of $3jm$ takes place after the calculation of the $6j$ and requires a further set of choices related to the subgroup chain to be made. The primitive $3jm$ must also be calculated. The non-primitive $3jm$ can then be calculated using the primitive $3jm$ and the $6j$ of group and subgroup.

The majority of this thesis is concerned with the calculation of $6j$ . Since the symmetry matrices are defined with respect to the $3jm$ we also discuss these symbols and find that one of the algorithms for calculating $6j$ is easily applied to them. Methods for calculating symmetrised powers of irreps, and the necessary extensions to the program SCHUR, are also discussed as this information is necessary for the calculation of $6j$.

In chapter 2 we review the properties of the $6j$, as this information is necessary background to the algorithms. These include the definitions of the symmetrised symbols and the equations that they obey. We extend the definition of the primitive irrep and define a new partial ordering for triads. The matrices that represent the symmetry relations are defined here in terms of the $3jm$. The form of these symmetry matrices and the values they may have are given in chapter 3, where we choose a form for the matrices of a triad with mixed symmetry.

We will show that it is possible to calculate recursively any non-primitive coupling, induction, recoupling or like process in terms of the primitive process. This algorithm is proven to be complete and valid for any multiplicity and represents a worthwhile improvement over the previous algorithm of Butler and Wybourne(1976). This reduces the problem of calculating such factors to one of calculating the primitive factors. The algorithm is discussed in chapter 4 and has been published in Searle and Butler(1988a).

We divide the primitive $6j$ into various classes in chapter 5. It is this classification that is the major advance of this thesis since we have previously had to solve an unknown primitive $6j$ by tediously searching for all the equations that involve the unknown. This is no longer necessary since we are able to prove that all the primitive $6j$ can be solved in terms of a single subclass which we shall call the core $6j$. Once a primitive non-core $6j$ has been classified we can then state directly the recursive equation that will solve it.

Chapter 6 discusses the number and occurrence of phase choices in the Racah-Wigner algebra. The basis $6j$ which we categorise here are shown to fix all the free phases and multiplicity separations occuring in the $6j$ part of the Racah-Wigner algebra. The basis $6j$ are a subset of the core $6j$.

In chapter 7 we discuss the problems that occur in the core $6j$ that prevent us from being able to give a complete algorithm for their solution. We prove that the core $6j$ can be completely solved for $SO_3$. The material in chapters 5 and 6, along with a slightly different but equivalent form of the proof for $SO_3$ has been published in Searle and Butler(1988b). We then discuss some of the types of problems that occur as we look at groups more complicated than $SO_3$, starting with the effects of multiplicity and irreps of the same power in the point groups and leading to the complications that occur in exceptional Lie groups like $G_2$ and $E_8$. The results of some $G_2$ and $E_8$ calculations are presented. The results for the point groups and the Lie groups are unpublished. The final group we consider in this chapter is a finite group, the K-metacyclic group of order 20, $K_{20}$, which has a complicated $6j$ algebra. This group has only one non-trivial irrep whose cube has a multiplicity of three, with a pair of mixed symmetry triads. We use the results of chapter 3 for mixed symmetry to completely solve the $6j$ for this group. This is one of the few calculations of $6j$ or $3jm$ with mixed symmetry (other examples are due to Zhang and Xiangzhu 1987, and Gao and Chen 1985), and is the first example known of strictly complex $6j$. The results for $K_{20}$ have been published with the necessary material from chapter 3 in Searle and Butler(1988c).

In chapter 8 we discuss the various data structures used to implement these ideas into a PASCAL program RACAH. This program is designed to actually calculate any $6j$ given the necessary selection rule information for the group. Examples are given for the data format the program uses for group information and the available commands are listed. The problem of getting data for $E_8$ to allow calculation of its $6j$ are also discussed here.

Throughout this thesis we intentionally use the term $6j$ to apply to a single $6j$ and a set of $6j$ symbols. Similar usage is made of the term $3jm$.

# Chapter 2

# Definitions

In this chapter we shall review the definitions and properties of the various generalised coupling factors. We shall assume familiarity with the standard $| jm >$ basis of $SO_3$ and with the concept of a representation space of a group. For more information on these see, for example, most modern books on quantum mechanics such as Messiah(1961).

The generalised coupling factors were first introduced by Racah(1949) in connection with the fractional parentage coefficients. Derome and Sharp (1965) considerd the properties of the symmetrised factors, the $6j$ and $3jm$. These results were reviewed by Butler(1975) and the properties of the symmetry relations were considered in detail. Bickerstaff(1981) also discusses these results.

It is to be noted that various similar but not always equivalent definitions have been used for the coupling and recoupling factors. In order to cover all possible compact groups we extend the definition of primitive irreps for a group and define a new partial ordering of triads.

## 2.1 The primitive representation

We say an irrep, $\epsilon_i$, is a primitive irrep, if it is part of the primitive representation, where the primitive representation is a selected faithful representation of the group. The primitive representation is usually chosen to be the smallest dimensional faithful representation, which in $SO_3$ is the irrep $\frac{1}{2}$. We remind readers here that a faithful representation has the property that its products generate all other representations (see Butler and Wybourne 1976, the Stone-Weyl theorem). In the double covering of $SO_3$ the vector, 1, is not faithful since its products do not contain any spinors. Since irreps are not always faithful the primitive representation may be reducible. We previously assumed that the primitive representation is a real irrep, $\epsilon$, or a complex pair $\epsilon + \epsilon^*$. This assumption is valid for the double covered point groups, the Lie groups $SU_n$, $SO_{2n+1}$, $Sp_{2n}$ and the exceptional groups. In $C_n$ we choose $\epsilon_1 = \frac{1}{2}$ and $\epsilon_2 = -\frac{1}{2}$, and in $SU_n$ $\epsilon_1 = \{1\}$ and $\epsilon_2 = \{1\}^*$. However for $SO_{2n}$ we need this new extended definition and choose $\epsilon_1 = [\frac{1}{2}]_+$ and $\epsilon_2 = [\frac{1}{2}]_-$, whilst for $D_2$ without spinors we choose $\epsilon_1 = 1$ and $\epsilon_2 = \tilde{1}$.

We are now able to define the power of any irrep $\lambda$ as the smallest value of $k$ for which

$$\left(\sum_i \epsilon_i\right)^k \supset \lambda$$

For example in $SO_3$

$$(\tfrac{1}{2})^3 \supset \tfrac{1}{2} + \tfrac{3}{2}$$

so $\frac{3}{2}$ is the only irrep of power 3. The power of any irrep can therefore be used to partially order the irreps.

The irreps of a group can behave as orthogonal or symplectic irreps even when they are complex. Such irreps are known as quasi-orthogonal or quasi-symplectic and the various types for the Lie groups are discussed in Butler and King(1974).

In later chapters we shall use the notation $\epsilon_1$, $\epsilon_2$ etc to refer to any of the primitive irreps. The dimension of an irrep $\lambda$ will be represented as $\mid \lambda \mid$.

## 2.2 Products and Triads

A triad $\lambda_1\lambda_2\lambda_3 r$ exists, if when we couple the three irreps the result, contains the scalar irrep, 0, at least r times. When $\lambda_1 \times \lambda_2 \times \lambda_3 \supset n\ 0$ there will be $n$ triads $\lambda_1\lambda_2\lambda_3$ with different multiplicity labels r. The product $\lambda_1 \times \lambda_2 \times \lambda_3 \supset n\ 0$ can alternatively be written as $\lambda_1 \times \lambda_2 \supset n\lambda_3^*$, which is the more general form of $j_1 \times j_2 \supset j_3$ in $SO_3$, in that the irrep $j_3$ may occur more than once. We know that in $SO_3$ the allowed values of $j_3$ are restricted so that

$$\mid j_1 - j_2 \mid \leq j_3 \leq j_1 + j_2 \tag{2.1}$$

This relation generalises to a restriction on the power of the irrep $\lambda_3$ in the general product

$$\mid p(\lambda_1) - p(\lambda_2) \mid \leq p(\lambda_3) \leq p(\lambda_1) + p(\lambda_2) \tag{2.2}$$

where there may be more than one $\lambda_3$ of a certain power that satisfies the product. A triad $\lambda_1\lambda_2\lambda_3$ is ordered when $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

In this thesis we shall define a triad in standard order as one where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\lambda_1\lambda_2\lambda_3 < \lambda_1^*\lambda_2^*\lambda_3^*$. The second of these conditions requires that $\lambda_1 \leq \lambda_1^*$. If $\lambda_1 = \lambda_1^*$ then one has that $\lambda_2 \leq \lambda_2^*$. Similarly if $\lambda_2$ is equal to its conjugate then the restriction $\lambda_3 \leq \lambda_3^*$ also applies. When irreps in the triad are of the same power this standard order is somewhat arbitrary since we have not defined a complete ordering of the irreps. For our purposes it does not matter how the irreps are fully ordered as long as complex conjugate pairs are kept consecutive.

We define a partial ordering on the triads by saying that $\lambda_1\lambda_2\lambda_3 < \mu_1\mu_2\mu_3$ when the triads are in standard order if $p(\lambda_2) + p(\lambda_3) < p(\mu_2) + p(\mu_3)$. This still leaves a significant number of triads that are equal so we shall extend our ordering by adding the condition that $\lambda_1\lambda_2\lambda_3 < \mu_1\mu_2\mu_3$ when $p(\lambda_2) + p(\lambda_3) = p(\mu_2) + p(\mu_3)$ if $p(\lambda_3) < p(\mu_3)$. This restriction implies that the triads with a more resticted range of powers in the product $\lambda_2 \times \lambda_3$, are less than those with a wider range of powers.

There is a further restriction that applies to most products of two irreps. In most groups we will consider the primitive rep is quasi-symplectic, and not all the powers in the range shown in (2.2) occur in the product. This situation occurs for the double covered point groups, the double covered $SO_n$ groups, $Sp_{2n}$, $SU_{2k}$, $F_4$ and $E_7$. In this case the minimum and maximum powers occur in the product but the powers that occur in the range differ by two, they are not integer steps of one. For example in $SO_3$ with the spinor as primitive

$$1 \times 1 \supset 0 + 1 + 2$$

In this example the product of two power two irreps produces irreps of power zero, two, and four, powers one and three do not occur. In all the usual groups that have quasi-symplectic primitive irreps we do not find any triads of the form $\epsilon\epsilon\epsilon$, $\epsilon\epsilon\epsilon^*$ or $2_p2_p\epsilon$ (where $2_p$ is any of the power two irreps). The only semi-simple Lie groups that do not have quasi-symplectic irreps are $SU_{2k+1}$, $G_2$, $E_6$ and $E_8$. All the groups we shall consider only contain quasi-orthogonal irreps in the product of two quasi-symplectic irreps. The exceptions to this are much rarer, although they do exist (see Butler 1975).

A triad which contains the scalar irrep is said to be a trivial triad. Any triad which contains the primitive rep but not the scalar irrep is said to be primitive. A triad that has $p(\lambda_1) = p(\lambda_2)+p(\lambda_3)$ is known as a stretched triad.

## 2.3 The coupling factors

The usual coupling coefficient, known variously as a vector coupling coefficient or a Clebsch-Gordan coefficient, arises when two spaces of $SO_3$, spanned by $\mid lm_l >$ and $\mid sm_s >$, are coupled (the tensor product of vector spaces are formed, then written in reduced form). The resulting states are labelled by a new basis $\mid (ls)jm_j >$, where the $j$ values are the reduction of the product $l \times s$. The coupling factor is just the matrix element of a unitary transformation between the bases.

$$\mid (ls)jm_j > = \sum_{m_l m_s} \mid lm_l sm_s >< lm_l sm_s \mid (ls)jm_j > \qquad (2.3)$$

The generalised coupling factor arises in essentially the same manner when we couple two bases , $\mid \lambda_1 l_1 >$ and $\mid \lambda_2 l_2 >$, for a general compact group and transform the coupled basis into the new basis $| r\lambda l >$. We must now include an extra label r to distinguish between possible multiple occurrences of the irrep $\lambda$ in the reduction of $\lambda_1 \times \lambda_2$.

$$| r\lambda l >= \sum_{l_1,l_2} | \lambda_1 l_1 \lambda_2 l_2 >< \lambda_1 l_1 \lambda_2 l_2 | r\lambda l > \tag{2.4}$$

In the angular momentum algebra of $SO_3$ the Condon and Shortley phase convention is used to fix our choice of phase for the coupling factors, but in general the unitary phase factors have an arbitrary phase choice in how they relate the two bases, so

$$< \lambda_1 l_1 \lambda_2 l_2 | r\lambda l >= \sum_{r'} < \lambda_1 l_1 \lambda_2 l_2 | r'\lambda l >^{alt} K(\lambda_1\lambda_2\lambda)_{rr'} \tag{2.5}$$

The unitary coupling factors defined here do not have very convenient properties under the interchange of the irreps $\lambda_i$ so the $3jm$ is defined as

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda^* \\ l_1 & l_2 & l \end{pmatrix}^r = \sum_{l'} | \lambda |^{-\frac{1}{2}} \begin{pmatrix} \lambda & \lambda^* \\ l & l' \end{pmatrix} < (\lambda_1\lambda_2) r\lambda l' | \lambda_1 l_1, \lambda_2 l_2 > \tag{2.6}$$

or in various similar but not necessarily equivalent ways. It will be seen that $3jm$ have simple symmetry properties.

The recoupling factor arises when we couple three basis states $| \lambda_1 l_1 >$, $| \lambda_2 l_2 >$ and $| \lambda_3 l_3 >$. Since we can treat $\lambda_1 \times \lambda_2 \times \lambda_3$ as either $(\lambda_1 \times \lambda_2) \times \lambda_3$ or $\lambda_1 \times (\lambda_2 \times \lambda_3)$ we have two possible basis states for the product reduction

$$| (\lambda_1\lambda_2) r_1 \lambda_{12}, \lambda_3, s\lambda l > \quad and \quad | \lambda_1(\lambda_2\lambda_3) r_2 \lambda_{23}, r\lambda l >$$

The recoupling factor is the unitary transformation between these two bases

$$\begin{aligned} | (\lambda_1\lambda_2) r_{12} \lambda_{12}, \lambda_3, r\lambda l > \quad = \quad & \sum_{r_{23}\lambda_{23}s} | \lambda_1(\lambda_2\lambda_3), r_{23}\lambda_{23}, s\lambda l > \\ & < \lambda_1(\lambda_2\lambda_3), r_{23}\lambda_{23}, s\lambda | (\lambda_1\lambda_2) r_{12}\lambda_{12}, \lambda_3, r\lambda > \end{aligned} \tag{2.7}$$

As with the coupling factors a more symmetrical symbol may be defined that is related to the recoupling factor, this is the $6j$ ( $\{\lambda\}$ and $M$ are defined in the next section)

$$\begin{aligned} & < (\lambda_1\lambda_2) r_{12}\lambda_{12}, \lambda_3, r_2\lambda | \lambda_1(\lambda_2\lambda_3), r_{23}\lambda_{23}, r_1\lambda >= \\ & \sum_{s_{12}s_2s_{23}s_1t_2t_{23}t_{12}} (| \lambda_{12} \| \lambda_{23} |)^{\frac{1}{2}} \{\lambda_2\} M((12)\lambda_{12}\lambda_3\lambda^*)_{s_2t_2} \\ & M((132)\lambda_2\lambda_3\lambda_{23}^*)_{s_{23}t_{23}} M((23)\lambda_1\lambda_2\lambda_{12}^*)_{s_{12}t_{12}} \\ & \begin{Bmatrix} \lambda_1 & \lambda_{23} & \lambda^* \\ \lambda_3^* & \lambda_{12} & \lambda_2 \end{Bmatrix}_{t_{12}t_{23}t_2s_1} \end{aligned} \tag{2.8}$$

which can be defined with various historical phases (see Butler 1975, equation 9.13). Unlike the $3jm$, the $6j$ or recoupling factor only contains irreps of the group concerned and is totally independent of the subgroups that occur in the basis. This is readily seen in the way that a $6j$ is related to the $3jm$,

$$
\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1 r_2 r_3 r_4} =
\sum_{i_1 i_2 i_3 i_4 i_5 i_6 i'_1 i'_2 i'_3 i'_4 i'_5 i'_6} \begin{pmatrix} \lambda_1 & \lambda_1^* \\ i_1 & i'_1 \end{pmatrix} \begin{pmatrix} \lambda_2 & \lambda_2^* \\ i_2 & i'_2 \end{pmatrix} \begin{pmatrix} \lambda_3 & \lambda_3^* \\ i_3 & i'_3 \end{pmatrix} \begin{pmatrix} \mu_1 & \mu_1^* \\ i_4 & i'_4 \end{pmatrix}
\begin{pmatrix} \mu_2 & \mu_2^* \\ i_5 & i'_5 \end{pmatrix} \begin{pmatrix} \mu_3 & \mu_3^* \\ i_6 & i'_6 \end{pmatrix} \begin{pmatrix} \lambda_1 & \mu_2^* & \mu_3 \\ i_1 & i'^*_5 & i_6 \end{pmatrix}^{r_1} \begin{pmatrix} \mu_4 & \lambda_2 & \mu_3^* \\ i_4 & i_2 & i'^*_6 \end{pmatrix}^{r_2}
\begin{pmatrix} \mu_1^* & \mu_2 & \lambda_3 \\ i'^*_4 & i_5 & i_3 \end{pmatrix}^{r_3} \begin{pmatrix} \lambda_1^* & \lambda_2^* & \lambda_3^* \\ i'^*_1 & i'^*_2 & i'^*_3 \end{pmatrix}^{r_4} \tag{2.9}
$$

since the summation is over all possible subgroup labels, leaving only the group irreps (see Derome and Sharp 1965, equation 5.1). The group triads in the four $3jm$ are the four triads that are contained in the $6j$.

In a similar manner to the triads, a $6j$ or $3jm$ that contains a scalar irrep is said to be trivial and a $6j$ or $3jm$ that contains a primitive irrep, but not a scalar, is said to be primitive.

## 2.4 The $K$,$A$ and $M$ matrices

The matrices that describe the symmetries of the $3jm$ and $6j$ symbols were introduced by Derome and Sharp(1965) and further discussed by Butler(1975) and Bickerstaff(1981). These matrices are $A$ for complex conjugation, $M$ for the various interchanges of a triad and $K$ (Derome and Sharp used $U$) for the phase freedom in the definition of the coupling (see 2.5).

We will start by considering the action of the $K$ matrix. This relates $3jm$ with different alternative coupling multiplicity choices via the unitary transformation

$$
\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^r_{alt} = \sum_{r'} K(\lambda_1\lambda_2\lambda_3)^r{}_{r'} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r'} \tag{2.10}
$$

This is equivalent to relating the various values for $3jm$ due to the different definitions of 2.6. The $K$ matrix therefore represents the freedom of choice we have in the $3jm$ symbol due to the coupling process (the $3jm$ also contains a further choice due to the branching multiplicity). We follow Derome(1966) in using this freedom to make particular choices for the $A$ and $M$ matrices in chapter 3.

The unitary $M$ matrix describes how the $3jm$ is affected by a column permutation (a reordering of the coupling, see Derome and Sharp 1965, equation

2.7), usually in the following manner

$$\begin{pmatrix} \lambda_a & \lambda_b & \lambda_c \\ i_a & i_b & i_c \end{pmatrix}^r_{alt} = \sum_{r'} M(\pi, \lambda_1\lambda_2\lambda_3)^r{}_{r'} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r'} \tag{2.11}$$

where (abc) is the permutation $\pi$ of (123). When M is diagonal, the diagonal elements are called 3j phases. The $2j$ symbol, $\{\lambda\}$, is equal to the $3j$ phase $\{\lambda^*\lambda 0\}$. The $K$ matrix relates alternative choices of $M$ matrix via

$$M'(\pi, \lambda_1\lambda_2\lambda_3) = K(\lambda_a\lambda_b\lambda_c)^\dagger M(\pi, \lambda_1\lambda_2\lambda_3)K(\lambda_1\lambda_2\lambda_3) \tag{2.12}$$

(see Derome 1966).

The $A$ matrix is the unitary transformation between the conjugated $3jm$ and the $3jm$ with conjugated irreps (see Derome and Sharp 1965, section 4)

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r*} = \sum_{r'i_1'i_2'i_3'} A(\lambda_1\lambda_2\lambda_3)_{rr'} \begin{pmatrix} \lambda_1 & \lambda_1^* \\ i_1 & i_1' \end{pmatrix} \begin{pmatrix} \lambda_2 & \lambda_2^* \\ i_2 & i_2' \end{pmatrix} \begin{pmatrix} \lambda_3 & \lambda_3^* \\ i_3 & i_3' \end{pmatrix} \begin{pmatrix} \lambda_1^* & \lambda_2^* & \lambda_3^* \\ i_1' & i_2' & i_3' \end{pmatrix}^{r'} \tag{2.13}$$

Various choices of this matrix are related by the $K$ and $M$ matrices in the following manner

$$A'(\lambda_1^*\lambda_2^*\lambda_3^*) = K(\lambda_1\lambda_2\lambda_3)^T A(\lambda_1^*\lambda_2^*\lambda_3^*)K(\lambda_1^*\lambda_2^*\lambda_3^*) \tag{2.14}$$

and

$$A(\lambda_a\lambda_b\lambda_c) = M(\pi^{-1}, \lambda_a^*\lambda_b^*\lambda_c^*)^T A(\lambda_1\lambda_2\lambda_3)M(\pi, \lambda_1\lambda_2\lambda_3)^\dagger \tag{2.15}$$

(see Butler 1975, section 8)

## 2.5 The properties of the $6j$

As a consequence of the above definitions and results one can derive (see Derome and Sharp 1965, section 5) the following symmetry properties for the $6j$. Since the $6j$ is related to a product of four $3jm$ it is easy to see that there will be four symmetry matrices in the relations, one for each of the four triads via equation (2. 9). Two alternative choices of coupling phase are related by

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r_4} = \sum_{r_1'r_2'r_3'r_4'} K(\lambda_1\mu_2^*\mu_3)^{r_1}{}_{r_1'} K(\mu_1\lambda_2\mu_3^*)^{r_2}{}_{r_2'} K(\mu_1^*\mu_2\lambda_3)^{r_3}{}_{r_3'} K(\lambda_1^*\lambda_2^*\lambda_3^*)^{r_4}{}_{r_4'} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1'r_2'r_3'r_4'} \tag{2.16}$$

The columns of a $6j$ are reordered via

$$\begin{Bmatrix} \lambda_a & \lambda_b & \lambda_c \\ \mu_a & \mu_b & \mu_c \end{Bmatrix}_{r_1 r_2 r_3 r_4} = \sum_{r_1' r_2' r_3' r_4'} M(\pi, \lambda_1 \mu_2^* \mu_3)^{r_1}_{r_1'} M(\pi, \mu_1 \lambda_2 \mu_3^*)^{r_2}_{r_2'} M(\pi, \mu_1^* \mu_2 \lambda_3)^{r_3}_{r_3'} M(\pi, \lambda_1^* \lambda_2^* \lambda_3^*)^{r_4}_{r_4'} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1' r_2' r_3' r_4'} \tag{2.17}$$

for $\pi$, an even permutation (M is usually +1 for $\pi$ even) and as

$$\begin{Bmatrix} \lambda_a & \lambda_b & \lambda_c \\ \mu_a & \mu_b & \mu_c \end{Bmatrix}^{\text{alt}}_{r_1 r_2 r_3 r_4} = \sum_{r_1' r_2' r_3' r_4'} \{\mu_1\}\{\mu_2\}\{\mu_3\} M(\pi, \lambda_1 \mu_2^* \mu_3)^{r_1}_{r_1'} M(\pi, \mu_1 \lambda_2 \mu_3^*)^{r_2}_{r_2'} M(\pi, \mu_1^* \mu_2 \lambda_3)^{r_3}_{r_3'} M(\pi, \lambda_1^* \lambda_2^* \lambda_3^*)^{r_4}_{r_4'} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1^* & \mu_2^* & \mu_3^* \end{Bmatrix}_{r_1' r_2' r_3' r_4'} \tag{2.18}$$

for an odd permutation $\pi$ of (123), see Butler (1975), equation 9.9. The complex conjugate of a $6j$ is

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}^*_{r_1 r_2 r_3 r_4} = \sum_{r_1' r_2' r_3' r_4'} A(\lambda_1 \mu_2^* \mu_3)_{r_1 r_1'} A(\mu_1 \lambda_2 \mu_3^*)_{r_2 r_2'} A(\mu_1^* \mu_2 \lambda_3)_{r_3 r_3'} A(\lambda_1^* \lambda_2^* \lambda_3^*)_{r_4 r_4'} \begin{Bmatrix} \lambda_1^* & \lambda_2^* & \lambda_3^* \\ \mu_1^* & \mu_2^* & \mu_3^* \end{Bmatrix}_{r_1' r_2' r_3' r_4'} \tag{2.19}$$

There is a final symmetry in the $6j$ that has no analogue in the $3jm$. This is the flip of a pair of columns

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1 r_2 r_3 r_4} = \begin{Bmatrix} \mu_1 & \mu_2^* & \lambda_3^* \\ \lambda_1 & \lambda_2^* & \mu_3^* \end{Bmatrix}_{r_1 r_2 r_3 r_4} \tag{2.20}$$

$$= \begin{Bmatrix} \lambda_1^* & \mu_2 & \mu_3^* \\ \mu_1^* & \lambda_2 & \lambda_3^* \end{Bmatrix}_{r_1 r_2 r_3 r_4} \tag{2.21}$$

$$= \begin{Bmatrix} \mu_1^* & \lambda_2^* & \mu_3 \\ \lambda_1^* & \mu_2 & \lambda_3^* \end{Bmatrix}_{r_1 r_2 r_3 r_4} \tag{2.22}$$

There are three relations between $6j$ that are not related by symmetry. These are the orthonormality equation

$$\sum_{\nu r_1 r_2} | \lambda_3 || \nu | \left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \nu \end{matrix} \right\}^*_{r_1 r_2 r_3 r_4} \left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3' \\ \mu_1 & \mu_2 & \nu \end{matrix} \right\}_{r_1 r_2 r_3' r_4'} = \delta_{\lambda_3 \lambda_3'} \delta_{r_3 r_3'} \delta_{r_4 r_4'} \tag{2.23}$$

the Racah backcoupling equation (where $\{\lambda\alpha\beta\} \equiv M((12)\lambda\alpha\beta)$ )

$$\left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{matrix} \right\}_{r_1 r_2 r_3 r_4} = \sum_{\nu r r'} | \nu | \{\mu_2\}\{\mu_1 \lambda_2 \mu_3^* r_2\}\{\lambda_1 \lambda_2 \lambda_3 r_4\}\{\mu_1 \lambda_1 \nu^* r'\} \left\{ \begin{matrix} \lambda_2 & \lambda_1 & \lambda_3 \\ \mu_1 & \mu_2 & \nu \end{matrix} \right\}_{r r' r_3 r_4} \left\{ \begin{matrix} \lambda_1 & \mu_1 & \nu \\ \lambda_2 & \mu_2 & \mu_3 \end{matrix} \right\}_{r_1 r_2 r r'} \tag{2.24}$$

and the Biedenharn-Elliott equation

$$\sum_r \left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{matrix} \right\}_{r_1 r_2 r_3 r} \left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \nu_1 & \nu_2 & \nu_3 \end{matrix} \right\}^*_{s_1 s_2 s_3 r} = \sum_{\rho t_1 t_2 t_3} | \lambda | \{\lambda_1\}\{\nu_1\}\{(123)\lambda_1 \nu_2^* \nu_3\}_{s_1 s_1'}\{(132)\nu_1 \lambda_2 \nu_3^*\}_{s_2 s_2'} \{(13)\lambda_1 \mu_2^* \mu_3\}_{r_1 r_1'}\{(23)\mu_1 \lambda_2 \mu_3^*\}_{r_2 r_2'}\{\mu_1^* \mu_2 \lambda_3 r_3\} \{\mu_1^* \nu_1 \lambda t_1\}\{\mu_2^* \nu_2 \lambda t_2\}\{\mu_3^* \nu_3 \lambda t_3\} \left\{ \begin{matrix} \nu_2 & \mu_2^* & \nu \\ \mu_3 & \nu_3 & \lambda_1^* \end{matrix} \right\}_{s_1 r_1 t_3 t_2} \left\{ \begin{matrix} \nu_3 & \mu_3^* & \nu \\ \mu_1 & \nu_1 & \lambda_2^* \end{matrix} \right\}_{s_2 r_2 t_1 t_3} \left\{ \begin{matrix} \nu_1 & \mu_1^* & \nu \\ \mu_2 & \nu_2 & \lambda_3^* \end{matrix} \right\}_{s_3 r_3 t_2 t_1} \tag{2.25}$$

We will also find it useful to know the equations between $3jm$. The $3jm$ obey two orthonormality equations

$$\sum_{\lambda r a_3} \frac{|\lambda|}{|\sigma_3|} \left( \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{matrix} \right)^{r*}_{s} \left( \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1'\sigma_1' & a_2'\sigma_2' & a_3\sigma_3 \end{matrix} \right)^{r}_{s'} = \delta_{ss'} \delta_{a_1 a_1'} \delta_{a_2 a_2'} \delta_{\sigma_1 \sigma_1'} \delta_{\sigma_2 \sigma_2'} \tag{2.26}$$

and

$$\sum_{a_1 a_2 \sigma_1 \sigma_2 s} \frac{|\lambda|}{|\sigma_3|} \left( \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{matrix} \right)^{r*}_{s} \left( \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3' \\ a_1\sigma_1 & a_2\sigma_2 & a_3'\sigma_3 \end{matrix} \right)^{r}_{s'} = \delta_{rr'} \delta_{a_3 a_3'} \delta_{\lambda\lambda'} \tag{2.27}$$

and the Wigner equation

$$\sum_r \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{pmatrix}_s^r \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1 r_2 r_3 r} =$$
$$\sum_{b_1 b_2 b_3 \rho_1 \rho_2 \rho_3 s_1 s_2 s_3} \begin{pmatrix} \mu_1 \\ b_1\rho_1 \end{pmatrix} \begin{pmatrix} \mu_2 \\ b_2\rho_2 \end{pmatrix} \begin{pmatrix} \mu_3 \\ b_3\rho_3 \end{pmatrix} \tag{2.28}$$
$$\begin{pmatrix} \lambda_1 & \mu_2^* & \mu_3 \\ a_1\sigma_1 & b_2\rho_2^* & b_3\rho_3 \end{pmatrix}_{s_1}^{r_1} \begin{pmatrix} \mu_1 & \lambda_2 & \mu_3^* \\ b_1\rho_1 & a_2\sigma_2 & b_3\rho_3^* \end{pmatrix}_{s_2}^{r_2}$$
$$\begin{pmatrix} \mu_1^* & \mu_2 & \lambda_3 \\ b_1\rho_1^* & b_2\rho_2 & a_3\sigma_3 \end{pmatrix}_{s_3}^{r_3} \begin{Bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \\ \rho_1 & \rho_2 & \rho_3 \end{Bmatrix}_{s_1 s_2 s_3 s}$$

The above equations for the $6j$ and $3jm$ were first given for an arbitrary compact group by Derome and Sharp(1965) (but in a different notation, the notation in these equations is that of Butler 1981).

# Chapter 3

# The Symmetry Matrices

We have seen in the previous chapter that there are two unitary matrices, $A$ and $M$, whose values determine the symmetry relations for the $6j$ and $3jm$, and there is a third unitary matrix $K$ which describes the freedom of choice. Once we have chosen $M$ the freedom in $K$ is partially restricted by equation (2.12). This still leaves us some freedom which we may use when we choose $A$. Most previous calculations have used the choice $A = I$ since this produces the convenient symmetry condition for $6j$

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}^*_{r_1r_2r_3r_4} = \begin{Bmatrix} \lambda_1^* & \lambda_2^* & \lambda_3^* \\ \mu_1^* & \mu_2^* & \mu_3^* \end{Bmatrix}_{r_1r_2r_3r_4} \tag{3.1}$$

This choice, however convenient, does not always produce symbols with real values, even if it can be shown that it is possible to always find a form in which all the values are real, as Sullivan(1983), Bickerstaff(1984), and Bickerstaff and Damhus(1985) have shown. In what follows we shall try to choose $A = I$, both for the convenience and for consistency with previous results.

## 3.1 Plethysms

The previous chapter was based on the result that the product of two irrep spaces can be decomposed in terms of irrep spaces of the group. When the product is the square of an irrep it can also be reduced in terms of irrep spaces of the group $S_2$. $S_2$ has two irreps [2], $[1^2]$ which are symmetric and antisymmetric respectively. This allows us to split the square of an irrep into symmetric and antisymmetric parts. This splitting can be found by evaluating the symmetrised squares of the irrep $\lambda$, $\lambda \otimes \{2\}$ and $\lambda \otimes \{1^2\}$, for the group. The concepts required for the evaluation of symmetrised powers are due to Littlewood,, but were first clearly enunciated by Wybourne(1970). The technique requires the irrep to be related to a Schur function, followed by the evaluation of a Schur function plethysm. An example of this using the program

SCHUR is discussed in section 8.8. In similar fashion we can split up the cube of an irrep with respect to the group $S_3$. $S_3$ has the irreps [3],[21] and [$1^3$] which are considered to be symmetric, of mixed symmetry and antisymmetric respectively. The splitting is achieved by evaluating $\lambda \otimes \{3\}$, $\lambda \otimes \{21\}$ and $\lambda \otimes \{1^3\}$. The mixed symmetry term, [21], needs special treatment since it is a two dimensional irrep.

## 3.2 *A* and *M*

Derome(1966) showed that the relation (2.12) allows relatively simple choices of the $M$ matrices to be made. He found that when all three irreps in the triad are not equivalent $M$ may be chosen to be diagonal, with diagonal entries $\pm 1$. This chooses $K(abc)$ with respect to $K(123)$. When $\lambda_1 = \lambda_2 \neq \lambda_3$ the fact that two irreps are identical somewhat restricts our choice. Derome found that we may again choose values of $\pm\delta_{rs}$ but we are not free to choose the sign as before. We must choose the diagonal element to be $+1$ if the irrep $\lambda_3$ occurs in the symmetric square of $\lambda_1$ and $-1$ if it occurs in the antisymmetric square, and therefore it is necessary to evaluate the appropriate plethysms. These results are also discussed by Butler(1975).

The case where $\lambda_1 = \lambda_2 = \lambda_3$ is the most complicated of all of the possibilities. We may again choose our matrix to be block diagonal on the symmetry type but three symmetry types may occur in the cube. As before we may choose $+\delta_{rs}$ or $-\delta_{rs}$ if the scalar occurs in the symmetric ($\lambda \otimes \{3\}$) or antisymmetric ($\lambda \otimes \{1^3\}$) part of the cube of $\lambda$ respectively. The interesting case occurs with those multiplicity values that occur in the mixed symmetry cube of $\lambda_1$, $\lambda \otimes \{21\}$. These cases do not occur in the point groups or the Lie groups $SO_3$ and $SU_3$ but do occur in some of the finite groups and in the rest of the Lie groups. The irrep [21] of $S_3$ is two dimensional so the triads with this symmetry occur in pairs and hence the block with this symmetry is itself composed of two dimensional blocks. So the matrix $M$ will have the form

$$M(\pi, \lambda\lambda\lambda) = \begin{pmatrix} I_{[3]} & & \\ & M_{[21]} & \\ & & -I_{[1^3]} \end{pmatrix} \tag{3.2}$$

as in Butler(1975). The dimension of $M_{[21]}$ is twice the multiplicity of the scalar in $\lambda \otimes \{21\}$. Derome(1966) solved this case to the extent of showing that the matrices are elements of $S_3$. Butler(1975) defined $M_{[21]}$ in terms of the real representation of [21] discussed in the next section. We shall use a different but related definition of the block diagonal $M_{[21]}$. The diagonal is composed of two dimensional blocks, where each block is the two dimensional representation matrix of [21] in $S_3$ for the appropriate permutation.

All the cases where the diagonal elements are $\pm 1$ have been considered before and it has been shown (Butler 1975, Bickerstaff 1981) that it is possible to choose $A = I$ for these cases. The two dimensional mixed symmetry case is more complex and will be considered next.

## 3.3 Mixed symmetry and the choice of $M$

The usual matrices for the two dimensional representation [21] of $S_3$ consist of real orthogonal matrices with the generators

$$(12) = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \quad and \quad (123) = \tfrac{1}{2}\begin{pmatrix} -1 & \sqrt{3} \\ -\sqrt{3} & -1 \end{pmatrix}$$

As a result, for a mixed symmetry triad pair $\lambda\lambda\lambda_1$ and $\lambda\lambda\lambda_2$, the $3jm$ with permuted columns is a linear combination of $3jm$ with unpermuted columns. In particular we have

$$\begin{pmatrix} \lambda & \lambda & \lambda \\ k & i & j \end{pmatrix}^1 = -\tfrac{1}{2}\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^1 + \tfrac{\sqrt{3}}{2}\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^2 \tag{3.3}$$

An alternative representation of this irrep has the second of these generators (the 3 cycle) diagonalised, giving

$$(12) = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \quad and \quad (123) = \begin{pmatrix} \omega^2 & \\ & \omega \end{pmatrix}$$

with $\omega = exp(2\pi\imath/3)$. Use of this matrix irrep would imply the use of complex $3jm$, but would avoid the occurrence of linear combinations, e.g.

$$\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^1 = \begin{pmatrix} \lambda & \lambda & \lambda \\ j & i & k \end{pmatrix}^2 = \omega^2\begin{pmatrix} \lambda & \lambda & \lambda \\ j & k & i \end{pmatrix}^1 \tag{3.4}$$

We wish to know if both of these choices are possible and the effect they will have upon the choice of $A$ matrix.

At this stage we impose the requirement that the product of two orthogonal or two symplectic irreps only contains orthogonal irreps. This is a restriction that is satisfied by all triads of all the classical Lie groups, point groups, symmetric groups and many finite groups (Butler 1975). With this restriction it has been shown (Butler 1975 and Bickerstaff 1981) that the $A$ matrix is a symmetric unitary matrix which is block diagonal on symmetry type. We would like to be able to choose the $A$ matrix to be the identity, as this is consistent with most previous worker's choices.

If the real choice of the $M$ matrix is made for the mixed symmetry case then equation (2.15) shows that the $A$ matrix must be a multiple of $I$. However, when we apply the generators of the complex choice of the $M$ matrix to $A$ we find

$$A(\lambda\lambda\lambda) = \begin{pmatrix} \omega & \\ & \omega^2 \end{pmatrix} A(\lambda\lambda\lambda) \begin{pmatrix} \omega & \\ & \omega^2 \end{pmatrix} \tag{3.5}$$

which requires that the diagonal elements of $A$ are zero. The other generator then fixes $A$ as a multiple of

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$$

This skew diagonal matrix relates one conjugated symbol to the symbol for the other multiplicity in the pair, e.g.

$$\begin{pmatrix} \lambda & \lambda & \lambda \\ j & i & k \end{pmatrix}^{1*} = \sum_{i'j'k'} \begin{pmatrix} \lambda & \lambda^* \\ i & i' \end{pmatrix} \begin{pmatrix} \lambda & \lambda^* \\ j & j' \end{pmatrix} \begin{pmatrix} \lambda & \lambda^* \\ k & k' \end{pmatrix} \begin{pmatrix} \lambda^* & \lambda^* & \lambda^* \\ i' & j' & k' \end{pmatrix}^2 \tag{3.6}$$

We will choose $A = I$ and will use the real set of permutation matrices. This keeps our convenient choice of $A$ which allows it to effectively be ignored for the $6j$ symbols. It also prevents the permuted form of a $6j$ being complex if one of the forms can be chosen real. This also avoids the permutation symmetries requiring some of the $6j$ values to be complex in groups where all the values can be chosen real.

# Chapter 4

# The Non-Primitive Factors

The first algorithm to specify how to calculate a complete class of $6j$ (and $3jm$) for a general compact group was that by Butler and Wybourne(1976). This method showed how to calculate all non-primitive $6j$ for the group, and a variation of it allowed the calculation of all non-primitive $3jm$.

This previous method combined the orthonormality equation with the Biedenharn-Elliott equation (the Wigner equation for $3jm$) to produce an equation with a single unknown on the left hand side. This removed the usual sum on multiplicity that occurs in the original equation (see 2.25) and results in the equation

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1 r_2 r_3 r_4} = \sum_{\nu_1 \nu_2} |\nu_1 \| \nu_2 \| \nu_3| \{\lambda_1\}\{\mu_1\}\{\mu_1 \nu_1^* \epsilon_i t_1\}\{\mu_2 \nu_2^* \epsilon_i t_2\}\{\mu_3 \nu_3^* \epsilon_i t_3\}$$
$$\{(13)\lambda_1 \nu_2^* \nu_3\}_{s_1 s_1'} \{(23)\nu_1 \lambda_2 \nu_3^*\}_{s_2 s_2'} \{\nu_1^* \nu_2 \lambda_3 s_3\} \tag{4.1}$$
$$\{(123)\lambda_1 \mu_2^* \mu_3\}_{r_1 r_1'} \{(132)\mu_1 \lambda_2 \mu_3^*\}_{r_2 r_2'} \begin{Bmatrix} \mu_2 & \mu_3^* & \lambda_1^* \\ \nu_3 & \nu_2 & \epsilon_i \end{Bmatrix}_{t_2 t_3 s_1' r_1'}$$
$$\begin{Bmatrix} \mu_3 & \mu_1^* & \lambda_2^* \\ \nu_1 & \nu_3 & \epsilon_i \end{Bmatrix}_{t_3 t_1 s_2' r_2'} \begin{Bmatrix} \mu_1 & \mu_2^* & \lambda_3^* \\ \nu_2 & \nu_1 & \epsilon_i \end{Bmatrix}_{t_1 t_2 s_3 r_3} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \nu_1 & \nu_2 & \nu_3 \end{Bmatrix}_{s_1 s_2 s_3 r}$$

This equation is used recursively so that all the non-primitive terms that occur are solved in terms of primitive ones. The unfortunate consequence of this is the introduction of an extra summation on the right hand side of the equation, but over an irrep not just a multiplicity. This extra summation results in more terms being required to solve the $6j$ than would occur if the original equation had been used to solve a $6j$ with multiplicity one. If the n unknowns that occur due to the summation on r in equation (2.25) are treated as a related set the Biedenharn-Elliott equation may be used to produce sufficient linear equations that the set may be solved simultaneously. This improvement to the algorithm requires less terms than the previous method, and it will be described in the rest of this chapter.

## 4.1 Solution of non-primitive $6j$

Our first step in solving a non-primitive $6j$ is to relate the unknown, via symmetry transformations, to a non-primitive $6j$ which has its lowest triad in standard order in the top row. Such a $6j$ will be written in the form

$$\left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{matrix} \right\}_{r_1 r_2 r_3 r_4} \tag{4.2}$$

For example, this means that we would relate by symmetry the $SO_3$ $6j$

$$\left\{ \begin{matrix} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{3}{2} & 2 & 2 \end{matrix} \right\} \tag{4.3}$$

to the $6j$ for which we would solve

$$\left\{ \begin{matrix} 2 & \frac{3}{2} & \frac{3}{2} \\ \frac{5}{2} & 2 & 3 \end{matrix} \right\} \tag{4.4}$$

Due to our definition of the order of triads a consequence of $\lambda_1\lambda_2\lambda_3 r_4$ being the lowest triad is that $\lambda_3$ will be the smallest irrep in the $6j$. Since the Biedenharn-Elliott (see 2.25) equation has on its left hand side a summation over $r_4$ we will have n unknowns, where n is the number of possible values of $r_4$. This set of unknowns is multiplied by a set of coefficient $6j$, so to guarantee that we can solve our n unknowns we must be able to show that the coefficients will allow us to produce a set of n independent linear equations. We shall use primitive $6j$ for the coefficients and choose them to be of the form

$$\left\{ \begin{matrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3 & \epsilon_i & \nu \end{matrix} \right\}_{s_1 s_2 s_3 r_4} \tag{4.5}$$

where $\bar{\lambda}_3$ is any irrep such that $p(\bar{\lambda}_3) = p(\lambda_3) - 1$ and we will have a choice of values for $\nu$.

We know that the $6j$ symbols are related to the unitary recoupling factors via (2.8). The orthonormality equation (2.23) for the $6j$ is the equivalent equation for the symmetrised symbol that embodies the unitarity of the recoupling factor. The set of $6j$ in (4.5) formed by varying $\lambda_3 s_3 r_4$ as a row index and $\nu s_1 s_2$ as a column index forms a square matrix $S$, so the orthonormality equation can be written as $SB = I$, where $B$ contains the coefficients and dimensional factors. This implies that $B = S^{-1}$, where only non-singular square matrices have inverses, and any non-singular square matrix must have $\lambda_3 s_3 r_4$

independent rows. We have therefore established that our set of coefficient $6j$ form an independent set. Since we only require n independent equations, where n is related to $\lambda_3 s_3 r_4$ by varying $r_4$ and keeping the other labels fixed, we are looking for a subset of this matrix. As the rows of the matrix are independent we can easily extract the n independent equations we require to solve our unknown set. If we take n rows from the matrix, n of the columns must be independent for the full rows to be independent in the full matrix. It is worth remarking that these n columns can come from any of the columns in the matrix and therefore we have no control over the values of $\nu$ we may be required to use in producing our n independent equations.

It is fairly obvious that this method will work as long as the $6j$ that occur on the right hand side, and our coefficient $6j$, are not dependent upon knowing the ones we are looking for. With these coefficients the Biedenharn-Elliott equation is

$$\begin{aligned}\sum_{r_4}\left\{\begin{matrix}\lambda_1 & \lambda_2 & \lambda_3\\ \mu_1 & \mu_2 & \mu_3\end{matrix}\right\}_{r_1r_2r_3r_4}\left\{\begin{matrix}\lambda_1 & \lambda_2 & \lambda_3\\ \bar{\lambda}_3 & \epsilon_i & \nu\end{matrix}\right\}^*_{s_1s_2s_3r_4} &= \\ \sum_{\lambda t_1t_2t_3} \mid \lambda \mid \{\lambda_1\}\{\bar{\lambda}_3\}\{(123)\lambda_1\epsilon_i^*\nu\}_{s_1s_1'}\{(132)\bar{\lambda}_3\lambda_2\nu^*\}_{s_2s_2'} & \\ \{(13)\lambda_1\mu_2^*\mu_3\}_{r_1r_1'}\{(23)\mu_1\lambda_2\mu_3^*\}_{r_2r_2'}\{\mu_1^*\mu_2\lambda_3r_3\} & \qquad (4.6)\\ \{\mu_1^*\bar{\lambda}_3\lambda t_1\}\{\mu_2^*\epsilon_i\lambda t_2\}\{\mu_3^*\nu\lambda t_3\}\left\{\begin{matrix}\epsilon_i & \mu_2^* & \lambda\\ \mu_3 & \nu & \lambda_1^*\end{matrix}\right\}_{s_1r_1t_3t_2} & \\ \left\{\begin{matrix}\nu & \mu_3^* & \lambda\\ \mu_1 & \bar{\lambda}_3 & \lambda_2^*\end{matrix}\right\}_{s_2r_2t_1t_3}\left\{\begin{matrix}\bar{\lambda}_3 & \mu_1^* & \lambda\\ \mu_2 & \epsilon_i & \lambda_3^*\end{matrix}\right\}_{s_3r_3t_2t_1} & .\end{aligned}$$

We will be solving the primitive $6j$ in terms of the primitive ones so the coefficient $6j$ and the two primitive $6j$ on the right hand side do not require knowledge of the unknown. The third $6j$ that occurs is not primitive but contains the triad $\nu\lambda_3\bar{\lambda}_3$. The irrep $\bar{\lambda}_3$ is lower than any of the irreps in the unknown $6j$ so $\nu\lambda_3\bar{\lambda}_3$ must be lower than $\lambda_1\lambda_2\lambda_3$. Since this is a non-primitive $6j$ we solve it by the same method and will get $6j$ with an even smaller triad as a result. Each unknown $6j$ is therefore dependent upon non-primitive $6j$ lower in the recursive chain and hence are independent of the unknown as we require. We see that we have produced a recursive algorithm for the solution of these $6j$ by following this chain down. The chain terminates when we find that $p(\bar{\lambda}_3) = 1$, resulting in all the $6j$ on the right hand side being primitive.

Our recursion differs from the previous method in that previously the orthonormality equation had been used to shift the coefficient $6j$ to the right hand side of the equation. This produced a summation on all values of the irrep $\nu$ in the coefficient $6j$. In our method we are only using the minimum number of values of $\nu$ required to produce a set of simultaneous equations, so we do not require as many terms, and will often not have to use the largest values of $\nu$, usually resulting in even smaller irreps and less steps in the recursive algorithm. The summation on $\lambda$ will be restricted the most if the ordering of the $6j$ results in $\mu_2$ being the lowest irrep in the $6j$ after $\lambda_3$.

## 4.2 Example

To give an example of how our recursive algorithm works in practice we will consider the solution to our previous example (4.4) of $SO_3$. ( We use the group $SO_3$ since it is familiar to most physicists. Since twice the $j$ value of the irrep is its power in $SO_3$ the form of these equations effectively give the form of the solution in powers of irreps for any group with a symplectic primitive rep.) We choose the coefficient $6j$ to be

$$\left\{\begin{array}{ccc} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \tag{4.7}$$

and get the equation

$$\begin{aligned}
&\left\{\begin{array}{ccc} 2 & \frac{3}{2} & \frac{3}{2} \\ \frac{5}{2} & 2 & 3 \end{array}\right\}\left\{\begin{array}{ccc} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{3}{2} \end{array}\right\} = \\
&\quad -4\left\{\begin{array}{ccc} 3 & \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & 2 & 2 \end{array}\right\}\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{3}{2} & 3 \end{array}\right\}\left\{\begin{array}{ccc} \frac{5}{2} & 2 & \frac{3}{2} \\ \frac{1}{2} & 1 & \frac{3}{2} \end{array}\right\} \\
&\quad +6\left\{\begin{array}{ccc} 3 & 2 & 2 \\ \frac{1}{2} & \frac{5}{2} & \frac{3}{2} \end{array}\right\}\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{5}{2} & 3 \end{array}\right\}\left\{\begin{array}{ccc} \frac{5}{2} & 2 & \frac{5}{2} \\ \frac{1}{2} & 1 & \frac{5}{2} \end{array}\right\}
\end{aligned} \tag{4.8}$$

The unknown is dependent upon the two non-primitve $6j$

$$\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{3}{2} & 3 \end{array}\right\} \quad \text{and} \quad \left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{5}{2} & 3 \end{array}\right\}$$

both of which can be solved in terms of the primitive $6j$ by using the coefficient $6j$

$$\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{array}\right\} \tag{4.9}$$

This results in the two equations

$$\begin{aligned}
&\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{3}{2} & 3 \end{array}\right\}\left\{\begin{array}{ccc} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{array}\right\} = \\
&\quad 5\left\{\begin{array}{ccc} 3 & \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & 2 & 1 \end{array}\right\}\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 2 \end{array}\right\}\left\{\begin{array}{ccc} \frac{5}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{array}\right\}
\end{aligned} \tag{4.10}$$

and

$$
\begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{5}{2} & \frac{5}{2} & 3 \end{Bmatrix}
\begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{Bmatrix} =
5\begin{Bmatrix} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 2 \end{Bmatrix}
\begin{Bmatrix} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 2 \end{Bmatrix}
\begin{Bmatrix} \frac{5}{2} & \frac{5}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{Bmatrix}
-7\begin{Bmatrix} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 3 \end{Bmatrix}
\begin{Bmatrix} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 3 \end{Bmatrix}
\begin{Bmatrix} \frac{5}{2} & \frac{5}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 3 \end{Bmatrix}
\tag{4.11}
$$

which contain only primitive $6j$ on the right hand side.

## 4.3 Application to other symbols

Our method for solving non-primitive $6j$ symbols can also be applied to other coupling factors for a general compact group, just as the previous algorithm could also be applied to $3jm$ symbols.

To apply it to the Wigner equation for $3jm$ we relate the unknown $3jm$ to a $3jm$ whose group triad is in standard order. We can then use the same coefficient $6j$ that we used for the Biedenharn-Elliott equation, to produce the required number of independent linear equations for the set of unknown $3jm$. The result is

$$
\sum_{r} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{pmatrix}^{r_4}_{s_4} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3 & \epsilon_i & \nu \end{Bmatrix}^{*}_{r_1r_2r_3r_4} =
\sum_{b_1b_2b_3\rho_1\rho_2\rho_3s_1s_2s_3} \begin{pmatrix} \mu_1 \\ b_1\sigma_1 \end{pmatrix} \begin{pmatrix} \mu_2 \\ b_2\sigma_2 \end{pmatrix} \begin{pmatrix} \mu_3 \\ b_3\sigma_3 \end{pmatrix}
\begin{pmatrix} \lambda_1 & \epsilon_i^* & \nu \\ a_1\sigma_1 & b_2\rho_2^* & b_3\rho_3 \end{pmatrix}^{r_1}_{s_1} \begin{pmatrix} \bar{\lambda}_3 & \lambda_2 & \nu^* \\ b_1\rho_1 & a_2\sigma_2 & b_3\rho_3^* \end{pmatrix}^{r_2}_{s_2}
\begin{pmatrix} \bar{\lambda}_3^* & \epsilon_i & \lambda_3 \\ b_1\rho_1^* & b_2\rho_2 & a_3\sigma_3 \end{pmatrix}^{r_3}_{s_3} \begin{Bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \\ \rho_1 & \rho_2 & \rho_3 \end{Bmatrix}_{s_1s_2s_3s}
\tag{4.12}
$$

As with the $6j$ we have two primitive $3jm$ on the right hand side of the equation, and a third non-primitive $3jm$ that involves a lower group triad which may be recursively solved for by this method.

This method can also be applied to other factors that obey equations similar to the Biedenharn-Elliott and Wigner equations, and which are unitary. An example of this are the induction and reinduction factors (Butler and Haase 1984). They show that the defining equation for the induction factor is very similar to the definition of the coupling factor, and the relation between four induction factors and a coupling factor is of the same form as the one that leads to the Wigner equation (Butler and Haase 1984, equation 5.7). Similarly the equation for the reinduction factors will have the same form as the recoupling factors (2.8) from which the Biedenharn-Elliott equation is derived.

So the method can be easily applied to such factors where the induction factor will act like the $3jm$ and the reinduction factor will take the place of the $6j$, with a primitive reinduction factor as the coefficient term of both equations.

# Chapter 5

# The Primitive $6j$

It has been possible to find algorithms which allow us to completely solve some of the primitive $6j$. The first step required by this process is to split the primitive $6j$ into four classes, each of which has a different method of solution. Having defined the various classes we then use the rest of this chapter to describe the method of solution for three of these classes. We shall leave the solution of the other class, the core $6j$, until chapter 7. The three classes we solve here are

1. Those $6j$ with a column of primitive irreps (5.10)

2. Those with at least one non-primitive triad and a primitive irrep in the sixth position (5.3)

3. Those with only one primitive irrep, where it is in the fourth position (5.1).

To be able to solve any of these cases recursively we must be able to show that the various classes of $6j$ only depend on the other classes of $6j$ of similar size (and ultimately the core $6j$) and $6j$ of the same class that are lower down the chain. This will require that in chapter 7 the core $6j$ only depend upon core $6j$ of the same size or lower $6j$ of the other primitive types.

## 5.1 Classification

By definition a primitive $6j$ contains at least one primitive irrep. Since each irrep is contained in two of the four triads in a $6j$, a primitive $6j$ has at least two primitive triads. We will consider the possible forms for $6j$ with two, three or four primitive triads separately, and then group these into various classes based on the method by which they are solved. In this chapter, where we use the irreps $\lambda'$, $\alpha'$, and $\beta'$, they must be within one of the power of $\lambda$, $\alpha$, and $\beta$ respectively due to the general restrictions on products. Also we ask that these labels, label irreps that are non-primitive. $n_p$, in the $6j$ will be used to represent the power, $n$, of that irrep, and $\epsilon_i$ is any one of the primitive irreps.

A $6j$ with two primitive triads can be related by the various symmetry relations to a $6j$ in one of the following forms

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \epsilon_1 & \beta' & \alpha' \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.1}$$

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \beta' & \epsilon_1 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.2}$$

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \alpha' & \lambda' & \epsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.3}$$

where the irreps in these $6j$ are ordered so that the largest non-primitive triad is in standard order in the top row. Therefore the forms are distinguished by the position of the primitive irrep in the bottom row. A consequence of the arbitrary ordering of irreps of the same power is that this does not totally distinguish the $6j$ in (5.1)-(5.3). When $p(\lambda) = p(\alpha)$ in (5.1) it is possible to use symmetry to relate this $6j$ to (5.2), since the standard order for a triad only requires that $\lambda \geq \alpha$. In a similar way (5.2) and (5.3) are related when $p(\alpha) = p(\beta)$, and all three forms are related if $p(\lambda) = p(\alpha) = p(\beta)$. We will therefore add the restriction that $p(\lambda) > p(\alpha)$ in (5.1) and $p(\alpha) > p(\beta)$ in (5.3) to completely distinguish the various forms (this effectively singles out 5.2 as will be seen later).

Since an example is always useful to illustrate what this means we will show how the definitions above effect the $6j$ of $SO_3$, starting with the $6j$ in its usual tabulated order.

$$\begin{Bmatrix} 4 & 4 & 1 \\ \frac{3}{2} & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \textit{ is related to } \begin{Bmatrix} 4 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 4 \end{Bmatrix}$$
$$\begin{Bmatrix} \frac{7}{2} & 3 & \frac{1}{2} \\ \frac{3}{2} & 2 & \frac{3}{2} \end{Bmatrix} \textit{ is related to } \begin{Bmatrix} \frac{7}{2} & \frac{5}{2} & 2 \\ \frac{3}{2} & \frac{1}{2} & 3 \end{Bmatrix}$$
$$\begin{Bmatrix} 4 & \frac{7}{2} & \frac{1}{2} \\ \frac{7}{2} & 3 & 2 \end{Bmatrix} \textit{ is related to } \begin{Bmatrix} \frac{7}{2} & \frac{7}{2} & 2 \\ 4 & 3 & \frac{1}{2} \end{Bmatrix}$$

In those cases where there was a possible choice of forms we find that

$$\begin{Bmatrix} \frac{7}{2} & 3 & \frac{1}{2} \\ 2 & \frac{3}{2} & 3 \end{Bmatrix} \textit{ is classified as } \begin{Bmatrix} 3 & 3 & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{7}{2} \end{Bmatrix} \textit{ not } \begin{Bmatrix} 3 & 3 & 2 \\ \frac{1}{2} & \frac{3}{2} & \frac{7}{2} \end{Bmatrix}$$
$$\begin{Bmatrix} \frac{7}{2} & 3 & \frac{1}{2} \\ \frac{5}{2} & 2 & \frac{5}{2} \end{Bmatrix} \textit{ is classified as } \begin{Bmatrix} 3 & \frac{5}{2} & \frac{5}{2} \\ 2 & \frac{1}{2} & \frac{7}{2} \end{Bmatrix} \textit{ not } \begin{Bmatrix} 3 & \frac{5}{2} & \frac{5}{2} \\ 2 & \frac{7}{2} & \frac{1}{2} \end{Bmatrix}$$

We can similarly relate a $6j$ with one non-primitive triad to a $6j$ in one of the following forms.

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \epsilon_1 & \epsilon_2 & \alpha' \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.4}$$

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta' & \epsilon_1 & \epsilon_2 \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.5}$$

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \epsilon_1 & \beta' & \epsilon_2 \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.6}$$

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \epsilon_1 & \epsilon_2 & \epsilon_3 \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.7}$$

As with the $6j$ with two non-primitive triads these $6j$ are not yet totally distinct, since the condition of standard order on the non-primitive triad can allow these forms to be related by symmetry. We therefore require that the usual condition on the non-primitive triad, $p(\lambda) \geq p(\alpha) \geq p(\beta)$, becomes the stricter $p(\lambda) > p(\alpha) > p(\beta)$ in (5.6), and $p(\lambda) > p(\beta)$ in (5.5).

Due to the restrictions on products, some of the $6j$ in (5.4)-(5.7) are very restricted, since the result of $\epsilon_1 \times \epsilon_2$ can be no larger than power 2. The consequence of this is that not all of the $6j$ in (5.4)-(5.6) are independent.

The first of the $6j$, (5.4), is relatively unrestricted and becomes

$$\left\{ \begin{array}{ccc} \lambda & \alpha & 2_p \\ \epsilon_1 & \epsilon_2 & \lambda' \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.8}$$

An example of this in $SO_3$ is that

$$\left\{ \begin{array}{ccc} 3 & \frac{5}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{array} \right\} \textit{ is related to } \left\{ \begin{array}{ccc} \frac{5}{2} & \frac{5}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 3 \end{array} \right\}$$

The $6j$ in (5.5) are very severely restricted by $p(\lambda) = 2$ and the requirement that the triad is ordered. This results in the only possible top triad being $2_p 2_p 2_p$. All the $6j$ with this top triad violate the constraint that $p(\lambda) > p(\beta)$ and can be related to (5.8), so for our purposes this form vanishes. A similar argument applies to (5.6) when $p(\alpha) = 2$ and $p(\beta) = 2$. Since $6j$ with $\lambda 2_p 2_p$ as the top triad can be related to (5.8), this form also vanishes. Finally there is only one possible type of $6j$ for (5.7) and that is

$$\left\{ \begin{array}{ccc} 2_p & 2_p & 2_p \\ \epsilon_1 & \epsilon_2 & \epsilon_3 \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{5.9}$$

Finally we have to consider the cases with no non-primitive triads. These are also arranged so that the largest triad (when there is one) is in the top row. The only case left where the $6j$ contains two primitive irreps is

$$\begin{Bmatrix} \lambda & \alpha & \epsilon_1 \\ \alpha' & \lambda' & \epsilon_2 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.10}$$

Those that have three or more primitive irreps heavily restrict the possible values of the other irreps that occur. For those groups where the primitive irrep is symplectic we only have these possible $6j$

$$\begin{Bmatrix} 2_p & \epsilon_1 & \epsilon_2 \\ 2_p & \epsilon_3 & \epsilon_4 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.11}$$

$$\begin{Bmatrix} 3_p & 2_p & \epsilon_1 \\ \epsilon_2 & 2_p & \epsilon_3 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.12}$$

If the primitive irrep is not symplectic then we must also include the following restricted cases

$$\begin{Bmatrix} 2_p & 2_p & \epsilon_1 \\ 2_p & \epsilon_2 & \epsilon_3 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.13}$$

$$\begin{Bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.14}$$

$$\begin{Bmatrix} 2_p & \epsilon_1 & \epsilon_2 \\ \epsilon_3 & \epsilon_4 & \epsilon_5 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.15}$$

$$\begin{Bmatrix} 2_p & 2_p & \epsilon_1 \\ \epsilon_2 & \epsilon_3 & \epsilon_4 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.16}$$

The pattern of conjugate irreps in the $6j$ result in the $6j$ with six primitive irreps vanishing when the primitive is a complex pair. The $6j$ does exist if the primitive rep is a single real orthogonal irrep. (It cannot exist if the primitive irrep is a pair of distinct real irreps since the triad $\epsilon_2\epsilon_1\epsilon_1$ implies that $\epsilon_1^2 \supset \epsilon_2$, which should then be a power two irrep). The only relatively unrestricted $6j$ for a non-symplectic primitive irrep is

$$\begin{Bmatrix} \lambda & \alpha & \epsilon_1 \\ \epsilon_2 & \epsilon_3 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.17}$$

Effectively all of the $6j$ with four primitive triads are in the form that they are usually listed in tables.

Now that we have described the possible types of primitive $6j$, we will group them into classes based upon the method of solution. We will group all $6j$ with a primitive irrep in the fifth position together with $6j$ with three or more primitive irreps and refer to these $6j$ as the core $6j$ (those in 5.2, 5.7, 5.8, 5.11-5.17). Any further discussion of these $6j$ will be postponed until chapter 7.

## 5.2 Solution of (5.10)

We shall proceed to prove that the Racah backcoupling equation can be used to solve the $6j$ in (5.10). The requirement that the top triad be the largest in the $6j$ means that there are restrictions on the values of the other irreps. Since $p(\lambda) \geq p(\alpha) > 1$ in the largest triad we find that if $p(\lambda) > p(\alpha)$ then $p(\lambda') < p(\lambda)$. The other possibility is that $p(\lambda) = p(\alpha)$ with $p(\lambda') \leq p(\lambda)$ and $p(\alpha') \leq p(\alpha)$. Using Racah backcoupling we find that

$$\begin{Bmatrix} \lambda & \alpha & \epsilon_2 \\ \alpha' & \lambda' & \epsilon_1 \end{Bmatrix}_{r_1 r_2 r_3 r_4} = \sum_{s_1 s_2 \rho = 0}^{2_p} \mid \rho \mid \{\lambda'\}\{\lambda\alpha\epsilon_2 r_4\}\{\alpha'\epsilon_2\epsilon_1^* r_2\}\{\epsilon_1\epsilon_2\rho s_2\} \begin{Bmatrix} \alpha & \epsilon_2 & \lambda \\ \epsilon_1 & \lambda' & \rho \end{Bmatrix}_{s_1 s_2 r_1 r_4} \begin{Bmatrix} \epsilon_2 & \epsilon_1 & \rho \\ \alpha & \lambda' & \alpha' \end{Bmatrix}_{r_3 r_2 s_1 s_2} \tag{5.18}$$

where the sum over $\rho$ is shown to range from power zero to two. Both of the $6j$ which the unknown depends upon are related to the $6j$ in (5.8), since they can be related to

$$\begin{Bmatrix} \alpha^* & \lambda' & \rho \\ \epsilon_1^* & \epsilon_2 & \lambda^* \end{Bmatrix}_{r_4 r_1 s_2 s_1} \quad and \quad \begin{Bmatrix} \alpha & \lambda'^* & \rho \\ \epsilon_2 & \epsilon_1^* & \alpha'^* \end{Bmatrix}_{r_2 r_3 s_2 s_1}$$

when we consider the constraints on $\lambda, \lambda', \alpha$ and $\alpha'$. These are core $6j$ which are either lower than the unknown, or of similar size (if $p(\lambda) = 2$ ), and which can be solved independently of the unknown.

To give an example of this method we will apply it to the $SO_3$ $6j$

$$\begin{Bmatrix} \frac{7}{2} & 3 & \frac{1}{2} \\ \frac{5}{2} & 3 & \frac{1}{2} \end{Bmatrix} \tag{5.19}$$

which is solved with the following equation

$$\begin{Bmatrix} \frac{7}{2} & 3 & \frac{1}{2} \\ \frac{5}{2} & 3 & \frac{1}{2} \end{Bmatrix} = -3 \begin{Bmatrix} \frac{7}{2} & \frac{5}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 3 \end{Bmatrix} \begin{Bmatrix} \frac{7}{2} & \frac{5}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 3 \end{Bmatrix} \tag{5.20}$$

## 5.3 Solution to (5.3)

We already require that $\lambda \geq \alpha > \beta$ in these $6j$. The requirement that $\lambda\alpha\beta$ is the largest non-primitive triad leads to the following conditions.

$$\begin{array}{lll} p(\lambda) = p(\alpha) & \quad requires & p(\lambda') \leq p(\lambda), p(\alpha') \leq p(\alpha) \\ p(\lambda) = p(\alpha) + 1 & and\ p(\alpha) = p(\alpha')\ requires & p(\lambda') \leq p(\alpha) \\ & and\ p(\alpha) < p(\alpha')\ requires & p(\lambda') < p(\lambda) \\ p(\lambda) > p(\alpha) & \quad requires & p(\alpha') \leq p(\alpha) \end{array}$$

These $6j$ can be completely solved by the Biedenharn-Elliott equation in the following manner

$$\sum_r \begin{Bmatrix} \lambda & \alpha & \beta \\ \alpha' & \lambda' & \epsilon_1 \end{Bmatrix}_{r_1r_2r_3r} \begin{Bmatrix} \lambda & \alpha & \beta \\ \bar{\beta} & \epsilon_2 & \nu \end{Bmatrix}^*_{s_1s_2s_3r} =$$
$$\sum_{t_1t_2t_3p(\rho)=p(\lambda')-1}^{p(\lambda')+1} |\rho| \{\lambda\}\{\bar{\beta}\}\{(123)\lambda\epsilon_2^*\nu\}_{s_1s_1'}\{(132)\bar{\beta}\alpha\nu^*\}_{s_2s_2'}$$
$$\{(13)\lambda\lambda'^*\epsilon_1\}_{r_1r_1'}\{(23)\alpha'\alpha\epsilon_1^*\}_{r_2r_2'}\{\alpha'^*\lambda'\beta r_3\}$$
$$\{\alpha'^*\bar{\beta}\rho t_1\}\{\lambda'^*\epsilon_2\rho t_2\}\{\epsilon_1^*\nu\rho t_3\}\begin{Bmatrix} \epsilon_2 & \lambda_2'^* & \rho \\ \epsilon_1 & \nu & \lambda^* \end{Bmatrix}_{s_1r_1t_3t_2} \tag{5.21}$$
$$\begin{Bmatrix} \nu & \epsilon_1^* & \rho \\ \alpha' & \bar{\beta} & \alpha^* \end{Bmatrix}_{s_2r_2t_1t_3} \begin{Bmatrix} \bar{\beta} & \alpha'^* & \rho \\ \lambda' & \epsilon_2 & \beta_3^* \end{Bmatrix}_{s_3r_3t_2t_1}$$

We note that the coefficient $6j$ used here is the same as that used in the previous chapter, where $\nu$ ranges over all possible values required to produce a set of independent equations. This coefficient $6j$ is now seen to be a core $6j$ of similar size to the unknown, and is independent since the core will be solved independently of these $6j$.

The first $6j$ on the right hand side is quite evidently related to (5.10). This $6j$ contains irreps lower than those in the unknown, so the $6j$ will definitely be independent of the unknown. To find the standard form of the other $6j$ in the equation we must make use of the constraints on the irreps, where we note that $p(\lambda') - 1 \leq p(\rho) \leq p(\lambda') + 1$ and $p(\lambda) - 1 \leq \nu \leq p(\lambda) + 1$. From the constraints we see that $\lambda, \alpha > \beta$ and $\lambda', \alpha \geq \beta$, so that $\bar{\beta}$ is less than all the irreps $\lambda, \lambda', \alpha$ and $\alpha'$. This also implies that $\nu \geq \beta$ so that $\nu > \bar{\beta}$. Since it is possible that $\rho < \lambda'$, we find that $\rho \geq \bar{\beta}$.

If we now look at the second $6j$ on the right hand side of the equation, we see that $\bar{\beta}$ occurs in both non-primitive triads, and if $p(\rho) = p(\bar{\beta})$ then $\nu\alpha\bar{\beta} > \alpha'\rho\bar{\beta}$. Therefore whatever the values of the other irreps in the $6j$ $\bar{\beta}$ will be the lowest irrep in the largest non-primitive triad. Since the primitive irrep is in the same column as $\bar{\beta}$ this $6j$ will be related to (5.3). This $6j$ is of the same form as the unknown but is independent of it since $\nu\alpha\bar{\beta} < \lambda\alpha\beta$.

The third $6j$ in the equation is a more complex since the form it relates to is dependent upon the actual values of the irreps that occur. The two non-primitive triads in the $6j$ are $\lambda'\alpha'\beta$ and $\rho\alpha'\bar{\beta}$. We know that $\bar{\beta} < \beta$ and $\alpha', \rho \geq \bar{\beta}$ so that $\lambda'\alpha'\beta$ is the largest triad. Since we also know that $\lambda', \alpha' \geq \beta$ it is always possible to arrange the $6j$ so that $\beta$ is in the third column. In the most common circumstances we have $\lambda' \geq \alpha'$ so that this $6j$ can be related to (5.2) with $\alpha$ in column two. But it is possible for $\alpha' > \lambda'$, where the third $6j$ would then have the primitive irrep in the first column and be related to (5.1). The next section will show that although the $6j$ in (5.1) are dependent on (5.3), the occurrence of $\bar{\beta}$ in the third $6j$ prevents it from being dependent upon the unknown.

Continuing with our $SO_3$ examples, we use the coefficient $6j$

$$\left\{\begin{array}{ccc} 3 & 3 & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{5}{2} \end{array}\right\} \tag{5.22}$$

to solve

$$\left\{\begin{array}{ccc} 3 & 3 & 2 \\ \frac{5}{2} & \frac{7}{2} & \frac{1}{2} \end{array}\right\} \tag{5.23}$$

resulting in the equation

$$\left\{\begin{array}{ccc} 3 & 3 & 2 \\ \frac{5}{2} & \frac{7}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & 3 & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{5}{2} \end{array}\right\} = \tag{5.24}$$
$$7\left\{\begin{array}{ccc} \frac{7}{2} & 3 & \frac{1}{2} \\ \frac{5}{2} & 3 & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ 3 & \frac{5}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} \frac{7}{2} & \frac{5}{2} & 2 \\ \frac{3}{2} & \frac{1}{2} & 3 \end{array}\right\}$$

We used the first of the $6j$ on the right hand side as the example in the previous section and the third $6j$ is core. We will also show how the second $6j$ is solved since this gives us an example of what happens when $\alpha' > \lambda'$. We use the coefficient $6j$

$$\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{array}\right\} \tag{5.25}$$

to arrive at

$$\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ 3 & \frac{5}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{array}\right\} = \tag{5.26}$$
$$5\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{1}{2} \\ 2 & \frac{5}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} \frac{5}{2} & \frac{5}{2} & 1 \\ 2 & 3 & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 2 \end{array}\right\}$$
$$-7\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{1}{2} \\ 3 & \frac{5}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & 3 & 1 \\ \frac{5}{2} & \frac{5}{2} & \frac{1}{2} \end{array}\right\}\left\{\begin{array}{ccc} 3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 3 \end{array}\right\}$$

where the third of these $6j$ is related to (5.1) as we predicted.

## 5.4 Solution to (5.1)

The values of the irreps in the $6j$ in (5.1) are also constrained by the condition that $p(\lambda) > p(\alpha) > p(\beta)$ which implies that $p(\alpha') < p(\alpha)$ if $p(\beta') > p(\beta)$ or $p(\alpha') \leq p(\alpha)$ if $p(\beta') = p(\beta)$. We shall again use the Racah backcoupling relation to solve this type of $6j$

$$\left\{\begin{array}{ccc}\lambda & \alpha & \beta \\ \epsilon_1 & \beta' & \alpha'\end{array}\right\}_{r_1r_2r_3r_4} = \sum_{s_1s_2p(\rho)=p(\lambda)-1}^{p(\lambda)+1} |\nu| \{\beta'\}\{\epsilon_1\alpha\alpha'^*r_2\}\{\lambda\alpha\beta r_4\}\{\epsilon_1\lambda^*\rho s_1\} \left\{\begin{array}{ccc}\alpha & \lambda & \beta \\ \epsilon_1 & \beta' & \rho\end{array}\right\}_{s_1s_2r_3r_4} \left\{\begin{array}{ccc}\lambda & \epsilon_1 & \rho \\ \alpha & \beta' & \alpha'\end{array}\right\}_{r_1r_2s_1s_2} \tag{5.27}$$

where $p(\lambda)-1 \leq p(\rho) \leq p(\lambda)+1$.

The two non-primitive triads in the first $6j$ are $\alpha\lambda\beta$ and $\alpha\beta'\rho$, where $\rho \geq \alpha$ and $\alpha \geq \beta'$. These two triads become $\lambda\alpha\beta$ and $\rho\alpha\beta'$ in standard form. It does not actually matter which of these triads is larger since both have $\alpha$, and therefore $\epsilon_i$, in the second column so that the $6j$ is related to (5.2), and is a core $6j$ of similar size to the unknown.

The second $6j$ has the two non-primitive triads $\lambda\alpha'\beta'$ and $\rho\alpha\beta'$. We already know that the second of these is in standard order. Since we require $\lambda > \alpha$, we know that $\lambda \geq \alpha'$ and if $\beta' \leq \beta$ then $\alpha' \geq \beta'$ and the first triad is also in standard order. This will mean that the second $6j$ is related to (5.3). When $\beta' > \beta$ we require that $\alpha' < \alpha$ making it possible to get $\beta' > \alpha$. In this case the second $6j$ becomes related to a core $6j$, and we will have no problems with it. In the previous section it was shown that it is possible for the $6j$ in (5.3) to depend upon these $6j$,and we have just shown that these $6j$ are related in turn to (5.3). This is not a problem since for those cases where it occurs, we have that $\beta' = \bar{\beta} < \beta$ and the $6j$ in (5.3) have $\lambda\alpha'\bar{\beta}$ as the largest triad. This triad is lower than the one in the unknown and therefore the $6j$ is independent.

Continuing with our examples we can now solve the third $6j$ on the right hand side of (5.26) as follows.

$$\left\{\begin{array}{ccc}3 & \frac{5}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & 3\end{array}\right\} = -6\left\{\begin{array}{ccc}3 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2}\end{array}\right\}\left\{\begin{array}{ccc}3 & 3 & 1 \\ \frac{5}{2} & \frac{5}{2} & \frac{1}{2}\end{array}\right\} + 8\left\{\begin{array}{ccc}3 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{7}{2}\end{array}\right\}\left\{\begin{array}{ccc}3 & 3 & 1 \\ \frac{5}{2} & \frac{5}{2} & \frac{1}{2}\end{array}\right\} \tag{5.28}$$

The first of the $6j$ is a core $6j$ of the same size as the unknown. As expected the second $6j$ is in the same form as the previous section, but as we predicted it involves 331 which is less than $3\frac{5}{2}\frac{3}{2}$ which occured in (5.25), so the solutions are independent.

# Chapter 6

# The Basis $6j$

Before considering the solution to the core $6j$ it is useful to consider the various freedoms that occur and their consequences. We expect to find orientation type choices in the algebra, similar to those in $3jm$, as well as the fundamental phase and multiplicity freedoms intrinsic to the Racah-Wigner algebra and represented by the $K$ matrix. The intrinsic freedom means that the phase and multiplicity separations must be chosen in a separate set that fixes the $K$ matrix. We shall find it useful to define a subset of the core $6j$, called the basis $6j$, where these freedoms are resolved. The definition we choose for the basis $6j$ is completely arbitrary, but will be seen to be convenient in terms of the algorithm for solving the core $6j$ presented in the next chapter. We shall see that the basis $6j$ significantly complicate the solution of the core $6j$.

## 6.1 The free phases

From our earlier discussion we know that it is possible to use the phase freedom of the coupling multiplicity space to get very simple choices for the permutation matrices $M(\pi)$. The choice of $M$ fixes eleven of the twelve forms of the $K$ matrix with respect to the remaining one (since the triad has twelve possible rearrangements). We shall therefore consider the $K$ matrix for $\lambda_1\lambda_2\lambda_3$ in standard form to be our free matrix. A few extra free phases have been found by Reid and Bulter(1980,1982) in the $3jm$ algebra when the branching rules for $G \supset H$ do not distinguish between irreps. This was shown to be related to orientation choices of the subgroup within the group. Since the reduction of the product of irreps of a group $G$ into irreps can be considered to be the branching $G \times G \supset G$, we considered it possible that such phases could also be found in the $6j$ algebra. We do in fact find some choices of this type, as the irreps $n$ and $\tilde{n}$ of $D_{2n}$ are in no way distinguished by the algebra. We shall consider the effect of this indistinguishability when we discuss the solution of the core $6j$.

## 6.2 Where the free phases occur

Two alternative choices of the coupling freedom in a $6j$ are related by four coupling phase factors as shown in (2.16). A trivial $6j$

$$\left\{ \begin{array}{ccc} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_2^* & \lambda_1 & 0 \end{array} \right\}$$

contains $K(\lambda_1, \lambda_2, \lambda_3)$ and $K(\lambda_1^*, \lambda_2^*, \lambda_3^*)$. The combination of these two matrices produce the identity, so that there is no freedom in the trivial $6j$. Therefore a trivial $6j$ cannot fix any of the $K$ matrices. There is also no freedom in the non-core $6j$ since in previous chapters we have shown that they may be completely solved in terms of the core $6j$. This means that the remaining freedom in the algebra, as contained in the $K$ matrices and possibly in extra orientation type choices, must be fixed within the set of core $6j$. The matrix, $K(\lambda\alpha\beta)$, can only be fixed when a single power of $K(\lambda\alpha\beta)$ remains in (2.16).

At certain points in the process of solving a set of $6j$ we will introduce a new triad. The $K$ matrix for this triad will not be fixed, so we will have a free phase. When attempting to solve a $6j$ with such a free phase we are only able to find equations involving its magnitude, and possibly symmetry relations which constrain the possible values of the phase. Choosing the phase of this $6j$, known as the basis $6j$ for the triad, then fixes the value of the $K$ matrix with respect to the other three $K$ matrices in (2.16).

When the new triad contains a new irrep $\lambda$, that is not the conjugate of a previous irrep, it is not quite this simple. Since any irrep is contained in two triads of the $6j$ we find that there are actually two new triads in the $6j$. The process of choosing the phase for such a basis $6j$ can only relate the $K$ matrix of one triad to the $K$ matrix of the other. Any other new triad containing this irrep $\lambda$ is then fixed with respect to one of the original matrices. We are never able to fix the phase of one of these two $K$ matrices within the Racah-Wigner algebra. We arbitrarily designate one of the two triads as still being free, and we call this triad the basis triad for the irrep $\lambda$. We usually chose the basis triad to be one of the stretched primitive triads containing $\lambda$ as the largest irrep. When the primitive irreps are symplectic, there is no basis triad for any of the $\epsilon_i$, otherwise the $n$ lowest non-stretched primitive triads can be considered to be the basis triads for the $n$ $\epsilon_i$.

## 6.3 Definition of the Basis $6j$

We shall choose the basis $6j$ for a triad $\lambda\alpha\beta$ (in standard form) to be

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta' & \epsilon_i & \lambda' \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{6.1}$$

where there is usually a certain amount of freedom in the choice of $\beta'$, $\epsilon_i$ and $\lambda'$. We aim to choose the irreps $\beta'$ and $\lambda'$ so that we have both $p(\beta') = p(\beta) - 1$ and $p(\lambda') = p(\lambda) - 1$. It is usually possible to satisfy these conditions for $\epsilon_i = \epsilon_1$ but there are occasions when other $\epsilon_i$ must be used instead. At the other extreme it may be that there are plenty of choices which allow us to satisfy these relations. In this case the actual $6j$ chosen to be basis is quite arbitrary, although it is usually convenient to choose the lowest values based upon the current ordering of irreps. The basis $6j$ may then be written as,

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \epsilon_1 & \bar{\lambda} \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{6.2}$$

which is a subset of (5.2).

There are two exceptions to (6.2), where it is impossible to find a $6j$ with phase freedom given our restrictions on $\beta'$ and $\lambda'$. The first exception occurs simply because the products in some groups are sufficiently restricted as to prevent such a choice. This is a common situation in the finite groups, and is often due to one of the irreps in the product being one dimensional. The only cure for this is to choose a larger value for $\beta'$ or $\lambda'$.

The second exception occurs when $p(\beta) = 1$, that is for all such triads $\lambda\alpha\epsilon_i$ except the first. Such triads occur when an irrep $\lambda$ has more than one primitive triad, a common occurrence in the Lie groups. In this case (6.2) would choose $p(\beta') = 0$. Such a choice does not contain the freedom required of a basis $6j$ since we have already shown that the resulting trivial $6j$ contains no freedom. This means that we must choose $p(\beta') = 1$ or 2 depending on which the group will allow.

If the group does not contain the triad $\epsilon_1\epsilon_2\epsilon_3$ then we are forced to choose $p(\beta') = 2$. This means that our basis $6j$ has the form

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \epsilon_1 \\ 2_p & \epsilon_2 & \bar{\lambda} \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{6.3}$$

where $p(\bar{\lambda}) = p(\alpha)$. This $6j$ is not in the standard form used in the last chapter and is related to a $6j$ in the form of (5.4) which may be written as,

$$\left\{ \begin{array}{ccc} \bar{\lambda} & \alpha & 2_p \\ \epsilon_1 & \epsilon_2 & \lambda \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{6.4}$$

where both $\lambda\alpha\epsilon_1$ and $\lambda\bar{\lambda}\epsilon_2$ are stretched primitives.

The alternative is that the group does contain the triad $\epsilon_1\epsilon_2\epsilon_3$ which usually allows us to find a basis $6j$ in the form

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \epsilon_1 \\ \epsilon_2 & \epsilon_3 & \bar{\lambda} \end{array} \right\}_{r_1 r_2 r_3 r_4} \tag{6.5}$$

where $\lambda\bar{\lambda}\epsilon_3$ is again a stretched primitive, but $\lambda\alpha\epsilon_1$ may not be. This $6j$ is in the same form as our earlier classes, and is a subset of (5.17).

When the triad $\lambda\alpha\beta$ has no multiplicity the choice of a single phase for the basis $6j$ is sufficient to fix $K(\lambda\alpha\beta)$. When the multiplicity n is greater than one, $K(\lambda\alpha\beta)$ is an n dimensional matrix. There are n basis $6j$ in the form of (6.2) for each multiplicity index. Depending on the available equations we will have to choose the phase of these $6j$, and we may in some cases have to also choose the magnitude separation. Fixing the values of these n $6j$ is not sufficient to fully specify the $K$ matrix, so extra choices must be made for other $6j$ containing the triad $\lambda\alpha\beta$ until all the elements of $K$ are fixed. The extra choices can be conveniently made for $6j$ of the same form as (6.2), making use of some of the other possible values of $\lambda'$. In the next chapter these $6j$ will be seen to occur in the same set of equations that contain $6j$ of the form (6.2).

# Chapter 7

# The Core $6j$

The problems that occur in solving the core $6j$ are essentially due to the basis $6j$. In the absence of multiplicity the basis $6j$ are only determined by quadratic equations, due to the essential phase freedom. Any set of core $6j$ we attempt to solve will most likely contain basis $6j$, producing a mixed set of linear and quadratic equations. This implies that we cannot get a proof of completeness of our algorithm of the type we have previously used since we do not have a set of linear equations. There are further complications when our first choice of basis $6j$ turns out to be zero, because we must then change our choice for the basis $6j$ until we find one that is non-zero. The result is that the orthonormality equation only solves most of the core $6j$. Even though no single equation is sufficient for a complete solution, testing of this algorithm on the various classes of point and Lie groups leads us to believe that we are not far from a complete algorithm.

The notation used for the point groups in this chapter is that of Butler(1981). Even though the group $SU_3$ is of major importance in physics, and the task that lead to this study was to extend the table of $SU_3$ $6j$ for C.Hamer, this group is not used as an example here. The special cases that occur in $SU_3$ are of the same type as the cases we discuss. We illustrate the special cases with groups simpler than $SU_3$.

The size of the set of core $6j$ relative to the number of primitive $6j$ depends on the distribution of the powers of irreps in the group. In finite groups with a small number of irreps of low power, the sets are identical, whilst in continuous groups the core set is significantly smaller.

Unlike the material in previous chapters, the algorithm in this chapter is incomplete. As a consequence the associated PASCAL routines are not fully debugged and the results are unpublished (M.D.Albrow's assistance in writing and testing various PASCAL routines is acknowledged, Albrow 1988).

## 7.1 Solving the core $6j$ - method

The starting point for solving the core $6j$ is the one equation we have not yet used to solve for $6j$ (although we have used its properties in the proof of completeness), the orthonormality equation. We have shown in section 4.1 that consideration of all possible values of irreps (and the associated multiplicities) in any column of a $6j$ gives a square non-singular matrix.

Given a $6j$

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \bar{\beta}^* & \epsilon_i & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \tag{7.1}$$

we will get such a non-singular matrix indexed by $\lambda' r_1 r_2$ and $\beta r_3 r_4$ , with relations between various matrix elements described by the orthonormality equation (2.24). Similarly for a $6j$

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \beta' & \epsilon_i & \bar{\lambda} \end{Bmatrix}_{r_1r_2r_3r_4} \tag{7.2}$$

we get a matrix labelled by $\lambda r_1 r_4$ and $\beta' r_2 r_3$ if we consider the values in the first column. There are two sets of equations that any one of these matrices can produce when we use the orthonormality equation. The first set in (7.1) arises from the sum on $\beta r_3 r_4$, and involves the dimension of $\beta$. The second set arises from a sum on the other index and is equivalent to the matrix equation on the transpose. These equations are distinct as long as the matrix is not hermitian, since the dimensions of $\beta$ and $\lambda'$ in the equations are then distinct. There are two similar sets of equations that can be produced from (7.2). If all the elements of these matrices were unknown, these equations would be insufficient to solve for them. This is not the case, in (7.1) there will be $6j$ with $\lambda' = \bar{\lambda}$, which are probably basis. There will also be rows in the matrix which are already known, since when $p(\beta) < p(\bar{\beta})$ the $6j$ is not in our classified form and will be solved by another matrix. Usually there are sufficient normality and orthogonality equations in the sets to solve the remaining unknown $6j$. If this was always true then the algorithm would be complete, but unfortunately these equations are not always sufficient.

The only other information available is the symmetry constraints on the value of the $6j$, and the Racah backcoupling and Biedenharn-Elliott equations. These last two equations relate the unknown $6j$ to a rearranged form of the unknown. This does not affect the $6j$ in (5.11)-(5.17), so the equations produce useful information. These equations do relate $6j$ in (5.2) to (5.1) or (5.3) which will usually result in the unknown depending upon itself further down the recursive chain, so they cannot be used in core $6j$ in (5.2).

When solving for the basis $6j$ we have to choose a phase. The reality of the irreps and of the free triad affects the choice of this phase. If the free triad and the $6j$ have no particular symmetry we may choose the phase, $e^{i\phi}$, to be

+1. For many groups all the irreps are real, and by (3.1) all the $6j$ are real, so the choice of $e^{i\phi}$ is restricted to $\pm 1$. When the triad is real, but the irreps are not, for example $\lambda^*\lambda\beta$ where $\beta = \beta^*$, the freedom is only a $\pm$ choice, and the symmetry of the $6j$ determines whether the choice is $\pm 1$ or $\pm i$. When the choice is free we usually choose it to be +1, although this can be altered to agree with historical choices.

We shall now give examples of how this method applies in practice, starting with the simpler cases and proceeding to the more complex ones.

## 7.2 The Angular Momentum group: $SO_3$

$SO_3$ is a continuous group with several special properties. All the irreps are real with exactly one irrep of each power and all products are multiplicity free. We find that the orthonormality equation is sufficient to solve all but one of the core $6j$ for $SO_3$. The equations we use for the core $6j$ are similar to the ones used by Butler(1976), except Butler(1976) uses orthonormality to solve for all primitive $6j$.

To solve for the $6j$

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda' \end{array} \right\} \tag{7.3}$$

we consider all possible values of the irreps in the third column of the $6j$. This produces a two dimensional matrix, since $\bar{\beta} \times \frac{1}{2}$ only contains the two unique irreps $\beta$ and $\bar{\beta} - \frac{1}{2}$.

$$\left( \begin{array}{cc} \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} & \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\} \\ \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} & \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\} \end{array} \right)$$

The two $6j$ in the top row are not in our standard form, and are related to

$$\left\{ \begin{array}{ccc} \lambda-\frac{1}{2} & \alpha & \bar{\beta} \\ \bar{\beta}-\frac{1}{2} & \frac{1}{2} & \lambda \end{array} \right\} \quad and \quad \left\{ \begin{array}{ccc} \lambda+\frac{1}{2} & \alpha & \bar{\beta} \\ \bar{\beta}-\frac{1}{2} & \frac{1}{2} & \lambda \end{array} \right\}$$

which are less than the unknown and would be solved by applying this method to the standard forms. This leaves us with two unknown $6j$ in the matrix, where the first $6j$ in the bottom row of the matrix is the basis $6j$ for $\lambda\alpha\beta$.

Applying the orthonormality equation with a sum on $\beta$ to this matrix gives

$$|\bar{\beta}-\tfrac{1}{2}| \left| \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \right|^2 + |\beta| \left| \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \right|^2 = \frac{1}{|\lambda-\frac{1}{2}|} \tag{7.4}$$

and

$$\begin{aligned} \mid \bar{\beta}-\tfrac{1}{2} \mid \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\} + & \\ \mid \beta \mid \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\} & = 0 \end{aligned} \tag{7.5}$$

which provide us with sufficient information to solve for the two unknowns. These two equations then give the result that, firstly

$$\left| \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \right|^2 = \frac{1}{\mid \beta \mid} \left[ \frac{1}{\mid \lambda-\frac{1}{2} \mid} - \mid \bar{\beta}-\tfrac{1}{2} \mid \left| \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \right|^2 \right] \tag{7.6}$$

The $6j$

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\}$$

is real since all the irreps are real, and the sign is chosen as $(-1)^{\lambda+\alpha+\beta}$ to agree with the usual choice. Secondly the other unknown is dependent on this basis $6j$ in the following manner

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\} = - \frac{\mid \bar{\beta}-\frac{1}{2} \mid \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\} \left\{ \begin{array}{ccc} \lambda & \alpha & \bar{\beta}-\frac{1}{2} \\ \bar{\beta} & \frac{1}{2} & \lambda+\frac{1}{2} \end{array} \right\}}{\mid \beta \mid \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \bar{\beta} & \frac{1}{2} & \lambda-\frac{1}{2} \end{array} \right\}} \tag{7.7}$$

The first column of the few $6j$ of the form

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta' & \frac{1}{2} & \bar{\lambda} \end{array} \right\}$$

also produces a two dimensional matrix with two unknowns and an orthogonality and normality equation. The matrix is

$$\left( \begin{array}{cc} \left\{ \begin{array}{ccc} \bar{\lambda}-\frac{1}{2} & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} & \left\{ \begin{array}{ccc} \bar{\lambda}-\frac{1}{2} & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} \\ \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} & \left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} \end{array} \right)$$

and produces the two solutions

$$\left\{ \begin{array}{ccc} \lambda & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} = \frac{1}{\mid \lambda \mid} \left[ \frac{1}{\mid \beta-\frac{1}{2} \mid} - \mid \bar{\lambda}-\frac{1}{2} \mid \left\{ \begin{array}{ccc} \bar{\lambda}-\frac{1}{2} & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array} \right\} \right] \tag{7.8}$$

$$\left\{\begin{array}{ccc} \lambda & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array}\right\} = -\frac{|\bar{\lambda}-\frac{1}{2}| \left\{\begin{array}{ccc} \bar{\lambda}-\frac{1}{2} & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array}\right\} \left\{\begin{array}{ccc} \bar{\lambda}-\frac{1}{2} & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array}\right\}}{|\lambda| \left\{\begin{array}{ccc} \lambda & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \bar{\lambda} \end{array}\right\}} \tag{7.9}$$

The one $SO_3$ $6j$ that will not solve by this method is

$$\left\{\begin{array}{ccc} \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \end{array}\right\}$$

This $6j$ is a core $6j$ of the form (5.12). The phase of this $6j$ is not free, but any column that is summed on only produces a one dimensional matrix with a normality equation or has the unknown in the top row. This $6j$ must be solved by the Racah backcoupling equation,

$$\begin{aligned} \left\{\begin{array}{ccc} \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \end{array}\right\} &= \left\{\begin{array}{ccc} 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \left\{\begin{array}{ccc} 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \\ &\quad - 3\left\{\begin{array}{ccc} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \left\{\begin{array}{ccc} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \end{aligned} \tag{7.10}$$

which is independent of the unknown.

## 7.3 Odd Dihedral groups: $D_{2n+1}$

Although the groups $D_{2n+1}$ are finite, their solution is slightly more complex than $SO_3$ since $D_{2n+1}$ contains two irreps of power 2, namely the pseudoscalar $\tilde{0}$ and the two dimensional irrep 1. Also the irreps of highest power are a complex pair, $\frac{n+1}{2}$ and $-(\frac{n+1}{2})$. This complex pair of irreps is indistinguishable in the coupling algebra, the labelling of the characters being quite arbitrary.

The occurrence of two irreps of the same power results in all the $6j$ of type (5.11) for the group being interrelated. This situation also occurs in the larger continuous groups and will be discussed more thoroughly in that context. Most of the other $6j$ of $D_{2n+1}$ solve in a manner identical to $SO_3$.

A very interesting situation occurs for the above complex pair of irreps. It illustrates a restriction on the phase choice that does not come from the symmetry relations of the $6j$. We shall use a $6j$ from $D_5$ to illustrate the problem. At first sight there are two possible basis $6j$ for the real triad $2\frac{3}{2}\frac{3}{2}$ which has a $\pm$ phase choice,

$$\left\{\begin{array}{ccc} 2 & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{3}{2} \end{array}\right\} \quad \textit{and} \quad \left\{\begin{array}{ccc} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{array}\right\}$$

Neither of these fully conform to the ideal in (6.2) since the products in these finite groups do not permit it. The first of these contains the free triad twice so it has no actual freedom associated with this triad and so this $6j$ cannot be used as the basis $6j$. The second is not related to its complex conjugate by any symmetry relation. However a sum on the third column produces the matrix

$$\begin{pmatrix} \begin{Bmatrix} 2 & \frac{3}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} & \begin{Bmatrix} 2 & \frac{3}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & -\frac{5}{2} \end{Bmatrix} \\ \begin{Bmatrix} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} & \begin{Bmatrix} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & -\frac{5}{2} \end{Bmatrix} \end{pmatrix}$$

From this it can be seen that the $6j$ in the second column are just the complex conjugates of those in the first column. The top row is known (by recursion) and the orthogonality equation requires that

$$\begin{Bmatrix} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix}^* = -\begin{Bmatrix} 2 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \tag{7.11}$$

so that this basis $6j$ must be pure imaginary. The normality equation gives us the magnitude of the $6j$ and the sign may be chosen $+$ to agree with Butler(1981), page 435.

## 7.4 Even Dihedral groups: $D_{2n}$

The irreps of the even dihedral groups are very similar to the odd, except that instead of a complex pair of irreps at the highest power we get the distinct irreps $n$ and $\tilde{n}$. Interestingly, these two irreps are indistinguishable in the product algebra just as are the complex irreps in $D_{2n+1}$, except that now the two irreps are not related. This leads to a situation very similar to the extra phases that occur in the $T \supset D_2$ $3jm$ (Reid and Butler 1980), where there is a choice of which one of the two dimensional irreps of $D_2$ an irrep of $T$ will be branched to. Each possible branching is related to a different orientation of the subgroup within the group. The extra freedom we find in $D_{2n}$ is of this type, if we treat the product as the branching $D_{2n} \times D_{2n} \supset D_{2n}$.

To illustrate what happens we shall consider the $D_6$ product $\frac{3}{2} \times \frac{3}{2} \supset 3 + \tilde{3}$. The basis $6j$ for $3\frac{3}{2}\frac{3}{2}$ and $\tilde{3}\frac{3}{2}\frac{3}{2}$ are

$$\begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \quad and \quad \begin{Bmatrix} \tilde{3} & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix}$$

These $6j$ are trivially solved by the general method since we get a one dimensional matrix by summing on all possible values of the irreps in the third column. The normality equation then gives the magnitude. However if we then solve the $6j$

$$\begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix}$$

we find that it belongs to the matrix

$$\begin{pmatrix} \begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} & \begin{Bmatrix} \tilde{3} & \frac{3}{2} & \frac{3}{2} \\ 1 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \\ \begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} & \begin{Bmatrix} \tilde{3} & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \end{pmatrix}$$

where we have already solved the top row. Neither of the $6j$ in the bottom row would normally be expected to contain any freedom since we have already solved the basis $6j$ for $3\frac{3}{2}\frac{3}{2}$, $\tilde{3}\frac{3}{2}\frac{3}{2}$ and the basis $6j$ for $\frac{5}{2}2\frac{3}{2}$ is

$$\begin{Bmatrix} \frac{5}{2} & 2 & \frac{3}{2} \\ 2 & \frac{1}{2} & 2 \end{Bmatrix}$$

This leaves us with only one independent orthogonality equation and a normality equation with which to solve two unknown $6j$, forcing us to make an unexpected phase choice. We find from the equations that

$$\begin{Bmatrix} \tilde{3} & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} = -\begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} \tag{7.12}$$

and

$$\begin{Bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ 2 & \frac{1}{2} & \frac{5}{2} \end{Bmatrix} = \pm\frac{1}{2} \tag{7.13}$$

So whichever root of the quadratic we choose for the $6j$ containing $3\frac{3}{2}\frac{3}{2}$ we will find that the other $6j$ will take the value of the other root.

## 7.5 Tetrahedral group: T

The tetrahedral group is a good example of some of the other special cases that can occur. The first example involves the complex irrep pair $\frac{3}{2}, -\frac{3}{2}$. When we try to solve the $6j$

$$\begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix}$$

we use column one to produce the matrix

$$\begin{pmatrix} \begin{Bmatrix} \frac{1}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} \frac{1}{2} & \frac{3}{2} & 1 \\ \frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} \frac{1}{2} & \frac{3}{2} & 1 \\ -\frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} \\ \begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ -\frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} \\ \begin{Bmatrix} -\frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} -\frac{3}{2} & \frac{3}{2} & 1 \\ \frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} & \begin{Bmatrix} -\frac{3}{2} & \frac{3}{2} & 1 \\ -\frac{3}{2} & \frac{1}{2} & 1 \end{Bmatrix} \end{pmatrix}$$

The first row and first column are known since the $6j$

$$\begin{Bmatrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{Bmatrix}_{000a}$$

solves in the general method using the third column. We have three orthogonality equations available to solve the other four $6j$ in the matrix. We would expect this to be insufficient. However the symmetry relations for the $6j$ involved make this matrix hermitian. This implies that there are only three independent unknowns for which we can solve.

The second useful special case is due to this being the first group we have considered that is not multiplicity free, since $1 \times 1 \supset (2)1$. The two triads $111_0$ and $111_1$ have different symmetry properties so it is possible to separate the magnitude of the two $6j$

$$\begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}_{0000} \quad and \quad \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}_{0001} \tag{7.14}$$

The second of these $6j$ turns out to be zero, and therefore has no useful freedom in its phase. As discussed in section 6.3 these two $6j$ are insufficient to completely fix the phase of the two dimensional unitary $K$ matrix. We must make a further choice of phase in the rest of the matrix containing the $6j$ with all possible values in column three,

$$\begin{pmatrix} \begin{Bmatrix} 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}_{0000} & \begin{Bmatrix} 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{Bmatrix}_{0000} & \begin{Bmatrix} 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} \end{Bmatrix}_{0000} \\ \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}_{0000} & \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{Bmatrix}_{0000} & \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} \end{Bmatrix}_{0000} \\ \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}_{0001} & \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{Bmatrix}_{0001} & \begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} \end{Bmatrix}_{0001} \end{pmatrix}$$

to completely fix $K$. Since the second of the $6j$ in (7.14) turns out to be zero, we can directly solve the $6j$

$$\begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{Bmatrix}_{0000}$$

using the orthogonality of column one to column two and then use the normality equation to find the magnitude of

$$\begin{Bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \end{Bmatrix}_{0001}$$

Fixing the phase of this $6j$ then completely fixes $K(111)_{rr'}$.

## 7.6 Octahedral group: O

In the octahedral group we encounter our first example of a multiplicity pair that is not separated by symmetry. The triad $\frac{3}{2}\frac{3}{2}1r$ has multiplicity two, with $M(\frac{3}{2}\frac{3}{2}1) = I$. The normality equation for the two basis $6j$

$$\left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0000} \quad and \quad \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0001}$$

gives us the sum of the magnitudes, and there is no equation which will separate them. We are therefore required to choose the magnitude separation as well as choosing the phase of each of the $6j$. Since this choice does not fully fix $K(\frac{3}{2}\frac{3}{2}1)_{rr'}$ we must make a further phase choice, which is easily done in the rest of the matrix (which uses the third column of the $6j$)

$$\begin{pmatrix} \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \tilde{1} \end{matrix}\right\}_{0000} \\ \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & \tilde{1} \end{matrix}\right\}_{0000} \\ \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0001} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{matrix}\right\}_{0001} & \left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & \tilde{1} \end{matrix}\right\}_{0001} \end{pmatrix}$$

If we choose

$$\left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}\right\}_{0001} = 0 \tag{7.15}$$

then the orthogonality equation produced by taking column one against column two, for

$$\left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{matrix}\right\}_{0000} \tag{7.16}$$

solves directly since it contains this $6j$ as the only unknown. It is then easy to use the normality equation which has only

$$\left\{\begin{matrix} \frac{3}{2} & \frac{3}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 2 \end{matrix}\right\}_{0001} \tag{7.17}$$

as an unknown, to finally fix $K$. This equation will have solutions of the type $\pm a$. If we do not choose either of the basis $6j$ to be zero then both the orthogonality and normality equations depend on (7.16) and (7.17). We must then solve the orthogonality equation and substitute the result into the normality equation. This produces a more general quadratic than previously with two distinct and equally valid solutions.

The octahedral group also has a pair of distinct irreps of power four, 2 and $\tilde{1}$. This gives a good example of the separation of irreps of the same power when both occur in a product. To solve the $6j$

$$\left\{\begin{matrix} \tilde{1} & \tilde{1} & \tilde{1} \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} \tag{7.18}$$

we produce the matrix

$$\begin{pmatrix} \left\{\begin{matrix} \tilde{1} & \tilde{1} & 1 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & 1 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0100} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & 1 \\ \frac{3}{2} & \frac{1}{2} & \frac{\tilde{1}}{2} \end{matrix}\right\}_{0000} \\ \left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0100} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{\tilde{1}}{2} \end{matrix}\right\}_{0000} \\ \left\{\begin{matrix} \tilde{1} & \tilde{1} & \tilde{1} \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & \tilde{1} \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0100} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & \tilde{1} \\ \frac{3}{2} & \frac{1}{2} & \frac{\tilde{1}}{2} \end{matrix}\right\}_{0000} \end{pmatrix}$$

This matrix contains the basis $6j$

$$\left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} \quad and \quad \left\{\begin{matrix} \tilde{1} & \tilde{1} & \tilde{1} \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000}$$

for the two triads $\tilde{1}\tilde{1}2$ and $\tilde{1}\tilde{1}\tilde{1}$. These $6j$ contain distinct triads and must therefore be distinct, but there is only one normality equation that contains both of these $6j$ with no other unknowns. This does not tell us how to separate their magnitudes. The problem is resolved by noting that since the irreps are the same power, we could have summed on the first column. This makes no difference for the triad $\tilde{1}\tilde{1}\tilde{1}$ but for $\tilde{1}\tilde{1}2$ it produces the two dimensional matrix

$$\begin{pmatrix} \left\{\begin{matrix} 1 & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} & \left\{\begin{matrix} 1 & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0100} \\ \left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0000} & \left\{\begin{matrix} \tilde{1} & \tilde{1} & 2 \\ \frac{3}{2} & \frac{1}{2} & \frac{3}{2} \end{matrix}\right\}_{0100} \end{pmatrix}$$

which allows us to solve the basis $6j$ for this triad. We can now use the previous normality equation to resolve the other basis $6j$ and proceed to completely solve the matrix.

## 7.7 The groups $G_2$ and $E_8$

Solving the $6j$ for either of these groups is essentially the same problem. Both of these groups involve a large number of irreps of each power which are real and non-symplectic. $G_2$ and $E_8$ are the only two semi-simple Lie groups (other than the orthogonal groups that are not doubly covered) for which the $6j$

$$\left\{ \begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{array} \right\} \tag{7.19}$$

exists. The physical relevance of $E_8$ is not currently known but the group $G_2$ is commonly used in the labelling of electrons in the $f$ orbitals of atoms (see Judd 1963). Symbols related to the $G_2$ $6j$ have also been calculated by Judd(1986), although few cases involving multiplicity were considered. In both these groups the primitive rep is the real orthogonal vector irrep, 1.

These groups give a worst case example of what is involved in solving the $6j$ in (5.11),(5.7) and (5.14)-(5.16) in a large continuous group, where the set of equations for solving for these core $6j$ seem to be inextricably interrelated. (Another worst case would be spin(n) for large n, as we have n irreps of power two in the product of the spinor, $\frac{1}{2}$, although we do not have the triad $\epsilon\epsilon\epsilon$). To solve $6j$ in these forms we make use of all equations that interrelate these forms. The orthonormality relations give equations of the form

$$\sum_{\lambda rs} |\lambda| \left\{ \begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \left\{ \begin{array}{ccc} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \lambda \end{array} \right\}^*_{rs00} = 0 \tag{7.20}$$

$$\sum_{\lambda rs} |\lambda| \left| \left\{ \begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \right|^2 = \frac{1}{|\epsilon|} \tag{7.21}$$

$$\sum_{\lambda rs} |\lambda| \left\{ \begin{array}{ccc} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \left\{ \begin{array}{ccc} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \lambda \end{array} \right\}^*_{rs00} = 0 \tag{7.22}$$

$$\sum_{\lambda rs} |\lambda| \left\{ \begin{array}{ccc} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \left\{ \begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda \end{array} \right\}^*_{rs00} = 0 \tag{7.23}$$

$$\sum_{\lambda rs} |\lambda| \left| \left\{ \begin{array}{ccc} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \right|^2 = \frac{1}{|2|} \tag{7.24}$$

$$\sum_{\lambda rs} |\lambda| \left| \left\{ \begin{array}{ccc} \epsilon & 2_p & \lambda \\ \epsilon & \epsilon & \epsilon \end{array} \right\}_{rs00} \right|^2 = \frac{1}{|\epsilon|} \tag{7.25}$$

$$\sum_{\lambda rs} |\lambda| \left| \left\{ \begin{array}{ccc} 2_p & 2_p & \lambda \\ \epsilon & \epsilon & \epsilon \end{array} \right\}_{rs00} \right|^2 = \frac{1}{|\epsilon|} \tag{7.26}$$

while Racah backcoupling gives

$$\left\{ \begin{array}{ccc} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \epsilon \end{array} \right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{\epsilon\epsilon 0\}\{\epsilon\epsilon\epsilon\}\{\epsilon\epsilon\lambda s\} \left\{ \begin{array}{ccc} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \lambda \end{array} \right\}_{rs00} \left\{ \begin{array}{ccc} \epsilon & \epsilon & \lambda \\ \epsilon & \epsilon & \epsilon \end{array} \right\}_{00rs} \tag{7.27}$$

$$\left\{\begin{matrix} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & 2_p \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{\epsilon\epsilon 0\}\{\epsilon\epsilon 2_p\}\{\epsilon\epsilon\lambda s\} \left\{\begin{matrix} \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} \epsilon & \epsilon & \lambda \\ \epsilon & \epsilon & 2_p \end{matrix}\right\}_{00rs} \tag{7.28}$$

$$\left\{\begin{matrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{\epsilon\epsilon\epsilon\}\{\epsilon\epsilon\epsilon\}\{\epsilon\epsilon\lambda s\} \left\{\begin{matrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} \epsilon & \epsilon & \lambda \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{00rs} \tag{7.29}$$

$$\left\{\begin{matrix} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{\epsilon\epsilon 2_p\}\{\epsilon\epsilon\epsilon\}\{\epsilon\epsilon\lambda s\} \left\{\begin{matrix} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} \epsilon & \epsilon & \lambda \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{00rs} \tag{7.30}$$

$$\left\{\begin{matrix} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & 2_p \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{\epsilon\epsilon 2_p\}\{\epsilon\epsilon 2_p\}\{\epsilon\epsilon\lambda s\} \left\{\begin{matrix} \epsilon & \epsilon & 2_p \\ \epsilon & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} \epsilon & \epsilon & \lambda \\ \epsilon & \epsilon & 2_p \end{matrix}\right\}_{00rs} \tag{7.31}$$

$$\left\{\begin{matrix} 2_p & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{2_p\epsilon\epsilon\}\{\epsilon\epsilon\epsilon\}\{\epsilon 2_p\lambda s\} \left\{\begin{matrix} \epsilon & 2_p & \epsilon \\ \epsilon & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} 2_p & \epsilon & \lambda \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{00rs} \tag{7.32}$$

$$\left\{\begin{matrix} 2_p & \epsilon & \epsilon \\ 2_p & \epsilon & \epsilon \end{matrix}\right\}_{0000} =$$

$$\sum_{\lambda rs} |\lambda| \{\epsilon\}\{2_p\epsilon\epsilon\}\{2_p\epsilon\epsilon\}\{2_p 2_p\lambda\} \left\{\begin{matrix} \epsilon & 2_p & \epsilon \\ 2_p & \epsilon & \lambda \end{matrix}\right\}_{rs00} \left\{\begin{matrix} 2_p & 2_p & \lambda \\ \epsilon & \epsilon & \epsilon \end{matrix}\right\}_{00rs} \tag{7.33}$$

The major difference between Racah backcoupling and orthonormality for these $6j$ is that the backcoupling equation produces independent equations by including the $3j$ for the triads. The set of unknown $6j$ in these equations involves only core $6j$, some of which are basis. There is sufficient information in these equations to simultaneously solve for all the $6j$ with no freedom and for the magnitude of the basis $6j$, since the trivial $6j$ are known. The algebraic program REDUCE was used to solve this set. The $6j$ of $E_8$ in table 7.2 are the minimum set of core $6j$ that can be solved, and come directly from the above equations. Due to the freedom in the $K$ matrix the multiplicity separations are not fully specified. The $6j$ of $G_2$ in table 7.1 go beyond this minimum set and include irreps of power three. The labels used for the irreps are from Wybourne and Bowick(1977). The biggest problem I faced when studying these groups was in finding the $3j$ phases for the triads (requiring the

Table 7.1: Some $6j$ of $G_2$

| | |
|---|---|
| **1 1 1** | |
| 1 1 1 0000 | $-1/2.7$ |
| **2 1 1** | |
| 1 1 1 0000 | $1/2.3.7$ |
| 2 1 1 0000 | $5/2.9.3.7$ |
| **21 1 1** | |
| 1 1 1 0000 | $1/2.7$ |
| 2 1 1 0000 | $1/2.3.7$ |
| 21 1 1 0000 | $0$ |
| **2 2 1** | |
| 1 1 1 0000 | $1/2.9$ |
| 1 1 2 0000 | $1/2.3.7$ |
| 2 1 1 0000 | $-1/2.9.3$ |
| **21 2 1** | |
| 1 1 1 0000 | $1/2.3\sqrt{7}$ |
| 1 2 1 0000 | $-1/2.9.3$ |
| **21 2 1** | |
| 1 1 2 0000 | $-5/2.9.7$ |
| 2 1 1 0000 | $1/2.9\sqrt{7}$ |
| 2 1 2 0000 | $-\sqrt{11}/2.9.3\sqrt{5.7}$ |
| 2 1 2 0100 | $2/9\sqrt{5.7}$ |
| 21 1 1 0000 | $-1/3.7$ |
| **2 2 2** | |
| 1 1 1 0000 | $\sqrt{5.11}/2.9.3\sqrt{7}$ |
| 1 1 1 0001 | $0$ |
| 2 1 1 0000 | $\sqrt{11}/2.9.3\sqrt{5.7}$ |
| 2 1 1 0001 | $4/9.3\sqrt{5.7}$ |
| **21 2 2** | |
| 1 1 1 0000 | $1/2.9$ |
| 2 1 1 0000 | $-1/2.3.7$ |
| 21 1 1 0000 | $2/9.7$ |
| **21 21 2** | |
| 1 1 1 0000 | $1/3.7$ |
| **21 21 21** | |
| 1 1 1 0000 | $1/2.7$ |
| **3 2 1** | |
| 1 2 1 0000 | $1/9.3$ |
| 1 1 2 0000 | $-1/9.7$ |
| 2 1 2 0000 | $\sqrt{5}/9.3\sqrt{7.11}$ |
| 2 1 2 0100 | $0$ |
| 21 1 2 0000 | $-1/9.7$ |
| **31 2 1** | |
| 1 2 1 0000 | $-1/2.9.3$ |
| 1 1 2 0000 | $-2/9.7$ |
| 2 1 2 0000 | $\sqrt{11}/2.9.3\sqrt{5.7}$ |
| 2 1 2 0100 | $-1/8.9\sqrt{5.7}$ |
| 21 1 2 0000 | $1/4.9.7$ |
| **31 21 1** | |
| 1 2 1 0000 | $1/2.3\sqrt{2.7}$ |
| 1 21 1 0000 | $1/4.7$ |

calculation of symmetrised powers). Discussion of these symmetrised powers is postponed to section 8.8

## 7.8 $K_{20}$

To test the calculation of mixed symmetry triads using the above results we looked for a small finite group that contained such a triad. The K metacyclic group of order 20, $K_{20}$, is the smallest such group. It has four one dimensional irreps (0,1,2 and 3) and one four dimensional irrep. All the irreps are quasi-orthogonal, even though a pair is complex ($2 = 3^*$). All the primitive $6j$ that occur are core and the irrep with the mixed symmetry triad is also the primitive rep. The triads $444_1$ and $444_2$ form the only mixed symmetry pair. The information on this group was taken from Biedenharn et al(1968). This class of groups is also discussed by Bovier et al (1981).

The one basis $6j$ that does not contain 444r is

$$\left\{ \begin{matrix} 2 & 2 & 1 \\ 4 & 4 & 4 \end{matrix} \right\}_{0000} \tag{7.34}$$

and is trivially solved by use of the normality relations. Those few non-basis $6j$ that do not contain any of the triads 444r can then be readily solved by the Racah backcoupling equation.

Table 7.2: Some $6j$ of $E_8$

| | |
|---|---|
| $\mathbf{21^7\ 21^7\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/8.2.31$ |
| $\mathbf{21\ 21^7\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $-1/8.5.31$ |
| $21\ 21^7\ 21^7\ 0000$ | $-3/8.5.5.5.31$ |
| $\mathbf{21\ 21\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/2.5.5.31$ |
| $\mathbf{21\ 21\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $9/2.5.5.5.31\sqrt{7}$ |
| $\mathbf{31^6\ 21^7\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $0$ |
| $21\ 21^7\ 21^7\ 0000$ | $1/8.5.5.7.31$ |
| $31^6\ 21^7\ 21^7\ 0000$ | $1/8.5.7.7.31$ |
| $\mathbf{31^6\ 21\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/4.5.31\sqrt{5.7}$ |
| $\mathbf{31^6\ 31^6\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/8.7.31\sqrt{5}$ |
| $\mathbf{31^6\ 21\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $\sqrt{19}/5.5.7.31\sqrt{2.5}$ |
| $\mathbf{31^6\ 31^6\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $\sqrt{431}/4.5.5\sqrt{2.29.11987}$ |
| $21^7\ 21^7\ 21^7\ 0001$ | $0$ |
| $\mathbf{31^6\ 31^6\ 31^6}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $11/4.5.7.7.31\sqrt{2}$ |
| $21^7\ 21^7\ 21^7\ 0001$ | $0$ |
| $\mathbf{42^7\ 21^7\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/8.2.3.5.31$ |
| $21\ 21^7\ 21^7\ 0000$ | $1/8.5.5.5.31$ |
| $31^6\ 21^7\ 21^7\ 0000$ | $1/4.3.5.5.7.31$ |
| $42^7\ 21^7\ 21^7\ 0000$ | $23/8.2.9.3.5.5.5.31$ |
| $\mathbf{42^7\ 31^6\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/8.3.5\sqrt{5.7.31}$ |
| $\mathbf{42^7\ 42^7\ 21^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/8.2.9.5.5$ |
| $\mathbf{42^7\ 21\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $\sqrt{19}/2.5.5.5.31\sqrt{7}$ |
| $\mathbf{42^7\ 31^6\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/4.5.5.7\sqrt{31}$ |
| $\mathbf{42^7\ 31^6\ 31^6}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $1/4.3.5.31\sqrt{7}$ |
| $21^7\ 21^7\ 21^7\ 0001$ | $1/8.5.5.7\sqrt{3.7}$ |
| $\mathbf{42^7\ 42^7\ 21}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $\sqrt{13}/8.3.5.5.5\sqrt{3.7}$ |
| $\mathbf{42^7\ 42^7\ 31^6}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $\sqrt{13}/4.9.5.5.7\sqrt{5}$ |
| $\mathbf{42^7\ 42^7\ 42^7}$ | |
| $21^7\ 21^7\ 21^7\ 0000$ | $2/9.3.5.5\sqrt{7.31}$ |
| $21^7\ 21^7\ 21^7\ 0001$ | $\sqrt{13.19}/8.2.9.5.5.5\sqrt{31}$ |

Solving for the large number of remaining $6j$ is more complicated since the column permutation of any $6j$ is related to a linear combination of some others (since even when a permutation does not seem to alter the irreps, it will still permute the multiplicity indices).

For example, a (23) interchange of

$$\left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0011}$$

is

$$\left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0101}$$

which by the symmetry relations is equal to

$$\sum_{ab} M(23,444)_{aa'} M(23,444)_{bb'} \left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{00a'b'} \tag{7.35}$$

since $M(23,444)_{0a} = \delta_{0a}$.

Newmarch(1983) shows how to treat the 24 possible symmetry rearrangements of a $6j$ as elements of the symmetric group $S_4$, in order to find the number of $6j$ that cannot be solved by symmetry. These results were used to work out the number of $6j$, for the triad $444_r$ of multiplicity three and symmetries [3] and [21], that all the other rearrangements could be related to. We chose the 14 independent $6j$ to be

$$\begin{array}{ccc}
\left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0000} & \left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0011} & \left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0012} \\
\left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0111} & \left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0112} & \left\{\begin{array}{ccc} 4 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{1111} \\
\left\{\begin{array}{ccc} 1 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0000} & \left\{\begin{array}{ccc} 1 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0010} & \left\{\begin{array}{ccc} 1 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0110} \\
\left\{\begin{array}{ccc} 1 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0120} & \left\{\begin{array}{ccc} 2 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0000} & \left\{\begin{array}{ccc} 2 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0010} \\
\left\{\begin{array}{ccc} 2 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0110} & \left\{\begin{array}{ccc} 2 & 4 & 4 \\ 4 & 4 & 4 \end{array}\right\}_{0120} &
\end{array}$$

The orthogonality and Racah backcoupling equations (with normality for the few basis $6j$) were used in similar fashion to that described for $G_2$. By substituting for all the dependent $6j$ in the equations we were then able to obtain sufficient independent equations to resolve the remaining $6j$. The Racah backcoupling equations give the necessary extra information because they include information on the symmetry of the triads. The simultaneous equations for the independent $6j$ were created and solved by the algebraic program REDUCE, as were the symmetry relations for the related $6j$. The results were put into the Biedenharn-Elliott equation as an independent check that the results were correct. The set of $6j$ in table 7.3 are in the same format as used in the tables of Butler(1981).

The $6j$ for this group, $K_{20}$, are the first set of $6j$ to be calculated that are strictly complex. It is impossible to find a different $K$ matrix that will give pure real or imaginary values for the complex values without producing strictly complex values for some of the other $6j$. A change of multiplicity separation for the mixed symmetry pair merely shuffles the various pure real, pure imaginary and complex values around without removing the complex ones.

Table 7.3: The $6j$ of $K_{20}$

| | |
|---|---|
| **0 0 0** | |
| 0 0 0 0000 | $+1$ |
| **4 4 0** | |
| 0 0 4 0000 | $+1/2$ |
| 4 4 0 0000 | $+1/4$ |
| **4 4 4** | |
| 4 4 0 0000 | $+1/4$ |
| 4 4 0 0001 | 0 |
| 4 4 0 0002 | 0 |
| 4 4 0 0011 | $+1/4$ |
| 4 4 0 0012 | 0 |
| 4 4 0 0022 | $-1/4$ |
| 4 4 4 0000 | $+1/6$ |
| 4 4 4 0001 | 0 |
| 4 4 4 0002 | 0 |
| 4 4 4 0010 | 0 |
| 4 4 4 0011 | $-1/12$ |
| 4 4 4 0012 | 0 |
| 4 4 4 0020 | 0 |
| 4 4 4 0021 | 0 |
| 4 4 4 0022 | $+1/12$ |
| 4 4 4 0100 | 0 |
| 4 4 4 0101 | $+1/24$ |
| 4 4 4 0102 | $+1/8\sqrt{3}$ |
| 4 4 4 0110 | $+1/24$ |
| 4 4 4 0111 | $+1/12\sqrt{2}$ |
| 4 4 4 0112 | $-1/4\sqrt{6}$ |
| 4 4 4 0120 | $-1/8\sqrt{3}$ |
| 4 4 4 0121 | $+1/4\sqrt{6}$ |
| 4 4 4 0122 | $+1/12\sqrt{2}$ |
| 4 4 4 0200 | 0 |
| 4 4 4 0201 | $+1/8\sqrt{3}$ |
| 4 4 4 0202 | $-1/24$ |
| 4 4 4 0210 | $-1/8\sqrt{3}$ |
| 4 4 4 0211 | 0 |
| 4 4 4 0212 | $-1/6\sqrt{2}$ |
| 4 4 4 0220 | $-1/24$ |
| 4 4 4 0221 | $-1/6\sqrt{2}$ |
| 4 4 4 0222 | 0 |
| 4 4 4 1000 | 0 |
| 4 4 4 1001 | $+1/24$ |
| 4 4 4 1002 | $-1/8\sqrt{3}$ |
| 4 4 4 1010 | $+1/24$ |
| 4 4 4 1011 | $+1/12\sqrt{2}$ |
| 4 4 4 1012 | $+1/4\sqrt{6}$ |
| 4 4 4 1020 | $+1/8\sqrt{3}$ |
| 4 4 4 1021 | $-1/4\sqrt{6}$ |
| 4 4 4 1022 | $+1/12\sqrt{2}$ |
| 4 4 4 1100 | $-1/12$ |
| 4 4 4 1101 | $+1/12\sqrt{2}$ |
| 4 4 4 1102 | 0 |
| 4 4 4 1110 | $+1/12\sqrt{2}$ |
| 4 4 4 1111 | $+1/8$ |
| 4 4 4 1112 | 0 |
| 4 4 4 1120 | 0 |
| 4 4 4 1121 | 0 |
| 4 4 4 1122 | $+1/24$ |
| 4 4 4 1200 | 0 |
| **4 4 4** | |
| 4 4 4 1201 | $-1/4\sqrt{6}$ |
| 4 4 4 1202 | $-1/6\sqrt{2}$ |
| 4 4 4 1210 | $+1/4\sqrt{6}$ |
| 4 4 4 1211 | 0 |
| 4 4 4 1212 | $+1/24$ |
| 4 4 4 1220 | $-1/6\sqrt{2}$ |
| 4 4 4 1221 | $+1/24$ |
| 4 4 4 1222 | 0 |
| 4 4 4 2000 | 0 |
| 4 4 4 2001 | $-1/8\sqrt{3}$ |
| 4 4 4 2002 | $-1/24$ |
| 4 4 4 2010 | $+1/8\sqrt{3}$ |
| 4 4 4 2011 | 0 |
| 4 4 4 2012 | $-1/6\sqrt{2}$ |
| 4 4 4 2020 | $-1/24$ |
| 4 4 4 2021 | $-1/6\sqrt{2}$ |
| 4 4 4 2022 | 0 |
| 4 4 4 2100 | 0 |
| 4 4 4 2101 | $+1/4\sqrt{6}$ |
| 4 4 4 2102 | $-1/6\sqrt{2}$ |
| 4 4 4 2110 | $-1/4\sqrt{6}$ |
| 4 4 4 2111 | 0 |
| 4 4 4 2112 | $+1/24$ |
| 4 4 4 2120 | $-1/6\sqrt{2}$ |
| 4 4 4 2121 | $+1/24$ |
| 4 4 4 2122 | 0 |
| 4 4 4 2200 | $+1/12$ |
| 4 4 4 2201 | $+1/12\sqrt{2}$ |
| 4 4 4 2202 | 0 |
| 4 4 4 2210 | $+1/12\sqrt{2}$ |
| 4 4 4 2211 | $+1/24$ |
| 4 4 4 2212 | 0 |
| 4 4 4 2220 | 0 |
| 4 4 4 2221 | 0 |
| 4 4 4 2222 | $+1/8$ |
| 1 4 4 0000 | $-1/12$ |
| 1 4 4 0001 | $+1/6\sqrt{2}$ |
| 1 4 4 0002 | $+1/2\sqrt{6}$ |
| 1 4 4 1000 | $+1/6\sqrt{2}$ |
| 1 4 4 1001 | $+5/24$ |
| 1 4 4 1002 | $-1/8\sqrt{3}$ |
| 1 4 4 2000 | $+1/2\sqrt{6}$ |
| 1 4 4 2001 | $-1/8\sqrt{3}$ |
| 1 4 4 2002 | $+1/8$ |
| 4 1 4 0000 | $-1/12$ |
| 4 1 4 0001 | $+1/6\sqrt{2}$ |
| 4 1 4 0002 | $-1/2\sqrt{6}$ |
| 4 1 4 0100 | $+1/6\sqrt{2}$ |
| 4 1 4 0101 | $+5/24$ |
| 4 1 4 0102 | $+1/8\sqrt{3}$ |
| 4 1 4 0200 | $-1/2\sqrt{6}$ |
| 4 1 4 0201 | $+1/8\sqrt{3}$ |
| 4 1 4 0202 | $+1/8$ |
| 4 4 1 0000 | $-1/12$ |
| 4 4 1 0001 | $-1/3\sqrt{2}$ |
| 4 4 1 0002 | 0 |
| 4 4 1 0010 | $-1/3\sqrt{2}$ |
| 4 4 1 0011 | $+1/12$ |
| 4 4 1 0012 | 0 |
| **4 4 4** | |
| 4 4 1 0020 | 0 |
| 4 4 1 0021 | 0 |
| 4 4 1 0022 | $+1/4$ |
| 2 4 4 0000 | $+1/12$ |
| 2 4 4 0001 | $(1+3i)/12\sqrt{2}$ |
| 2 4 4 0002 | $(1-i)/4\sqrt{6}$ |
| 2 4 4 1000 | $(1-3i)/12\sqrt{2}$ |
| 2 4 4 1001 | $+1/24$ |
| 2 4 4 1002 | $(1+2i)/8\sqrt{3}$ |
| 2 4 4 2000 | $(1+i)/4\sqrt{6}$ |
| 2 4 4 2001 | $(1-2i)/8\sqrt{3}$ |
| 2 4 4 2002 | $+1/8$ |
| 4 2 4 0000 | $+1/12$ |
| 4 2 4 0001 | $(1-3i)/12\sqrt{2}$ |
| 4 2 4 0002 | $-(1+i)/4\sqrt{6}$ |
| 4 2 4 0100 | $(1+3i)/12\sqrt{2}$ |
| 4 2 4 0101 | $+1/24$ |
| 4 2 4 0102 | $(-1+2i)/8\sqrt{3}$ |
| 4 2 4 0200 | $(-1+i)/4\sqrt{6}$ |
| 4 2 4 0201 | $-(1+2i)/8\sqrt{3}$ |
| 4 2 4 0202 | $+1/8$ |
| 4 4 2 0000 | $+1/12$ |
| 4 4 2 0001 | $-1/6\sqrt{2}$ |
| 4 4 2 0002 | $+i/2\sqrt{6}$ |
| 4 4 2 0010 | $-1/6\sqrt{2}$ |
| 4 4 2 0011 | $+1/6$ |
| 4 4 2 0012 | $+i/4\sqrt{3}$ |
| 4 4 2 0020 | $-i/2\sqrt{6}$ |
| 4 4 2 0021 | $-i/4\sqrt{3}$ |
| 4 4 2 0022 | 0 |
| **4 1 4** | |
| 4 4 4 0000 | $-1/12$ |
| 4 4 4 0010 | $+1/6\sqrt{2}$ |
| 4 4 4 0020 | $-1/2\sqrt{6}$ |
| 4 4 4 1000 | $+1/6\sqrt{2}$ |
| 4 4 4 1010 | $+5/24$ |
| 4 4 4 1020 | $+1/8\sqrt{3}$ |
| 4 4 4 2000 | $-1/2\sqrt{6}$ |
| 4 4 4 2010 | $+1/8\sqrt{3}$ |
| 4 4 4 2020 | $+1/8$ |
| **4 2 4** | |
| 4 4 4 0000 | $+1/12$ |
| 4 4 4 0010 | $(1+3i)/12\sqrt{2}$ |
| 4 4 4 0020 | $(-1+i)/4\sqrt{6}$ |
| 4 4 4 1000 | $(1-3i)/12\sqrt{2}$ |
| 4 4 4 1010 | $+1/24$ |
| 4 4 4 1020 | $-(1+2i)/8\sqrt{3}$ |
| 4 4 4 2000 | $-(1+i)/4\sqrt{6}$ |
| 4 4 4 2010 | $(-1+2i)/8\sqrt{3}$ |
| 4 4 4 2020 | $+1/8$ |
| **4 4 1** | |
| 4 4 4 0000 | $-1/12$ |
| 4 4 4 0100 | $-1/3\sqrt{2}$ |
| 4 4 4 0200 | 0 |
| 4 4 4 1000 | $-1/3\sqrt{2}$ |
| 4 4 4 1100 | $+1/12$ |

# 7.3 continued

**4 4 1**

| | |
|---|---|
| 4 4 4 1200 | 0 |
| 4 4 4 2000 | 0 |
| 4 4 4 2100 | 0 |
| 4 4 4 2200 | $+1/4$ |

**4 4 2**

| | |
|---|---|
| 4 4 4 0000 | $+1/12$ |
| 4 4 4 0100 | $-1/6\sqrt{2}$ |
| 4 4 4 0200 | $+i/2\sqrt{6}$ |
| 4 4 4 1000 | $-1/6\sqrt{2}$ |
| 4 4 4 1100 | $+1/6$ |
| 4 4 4 1200 | $+i/4\sqrt{3}$ |
| 4 4 4 2000 | $-i/2\sqrt{6}$ |
| 4 4 4 2100 | $-i/4\sqrt{3}$ |
| 4 4 4 2200 | 0 |

**1 4 4**

| | |
|---|---|
| 0 4 4 0000 | $+1/4$ |
| 4 1 0 0000 | $+1/2$ |
| 4 4 4 0000 | $-1/12$ |

**1 4 4**

| | |
|---|---|
| 4 4 4 0010 | $+1/6\sqrt{2}$ |
| 4 4 4 0020 | $+1/2\sqrt{6}$ |
| 4 4 4 0100 | $+1/6\sqrt{2}$ |
| 4 4 4 0110 | $+5/24$ |
| 4 4 4 0120 | $-1/8\sqrt{3}$ |
| 4 4 4 0200 | $+1/2\sqrt{6}$ |
| 4 4 4 0210 | $-1/8\sqrt{3}$ |
| 4 4 4 0220 | $+1/8$ |
| 1 4 4 0000 | $+1/4$ |

**1 1 0**

| | |
|---|---|
| 0 0 1 0000 | $+1$ |
| 1 1 0 0000 | $+1$ |

**2 4 4**

| | |
|---|---|
| 0 4 4 0000 | $-1/4$ |
| 4 2 0 0000 | $-1/2$ |
| 4 4 4 0000 | $+1/12$ |
| 4 4 4 0010 | $(1+3i)/12\sqrt{2}$ |
| 4 4 4 0020 | $(1-i)/4\sqrt{6}$ |

**2 4 4**

| | |
|---|---|
| 4 4 4 0100 | $(1-3i)/12\sqrt{2}$ |
| 4 4 4 0110 | $+1/24$ |
| 4 4 4 0120 | $(1+2i)/8\sqrt{3}$ |
| 4 4 4 0200 | $(1+i)/4\sqrt{6}$ |
| 4 4 4 0210 | $(1-2i)/8\sqrt{3}$ |
| 4 4 4 0220 | $+1/8$ |
| 1 4 4 0000 | $-1/4$ |
| 2 4 4 0000 | $+1/4$ |
| 3 4 4 0000 | $+1/4$ |

**2 2 1**

| | |
|---|---|
| 0 1 2 0000 | $+1$ |
| 4 4 4 0000 | $+1/2$ |
| 2 3 1 0000 | $+1$ |
| 3 2 0 0000 | $+1$ |

**3 2 0**

| | |
|---|---|
| 0 0 2 0000 | $+1$ |
| 3 3 0 0000 | $+1$ |

# Chapter 8

# Algorithms

The original version of RACAH was written in Burroughs ALGOL by P.H. Butler, with later modification by R.W.Haase and M.F.Reid. This version was used to produce the tables in Butler(1981) and Bickerstaff et al(1982). After this it was decided to reimplement the RACAH in essentially ISO PASCAL to make it more portable. This effort was also prompted by the desire to incorporate substantial improvements to the algorithms. I have been responsible for various improvements to the program and have written procedures to carry out the algorithms presented in previous chapters. The algorithms for testing the ideas in chapter 7 were written by M.D.Albrow. This chapter discusses some of the algorithms and data structures that are required, and the reasons for their use.

## 8.1 C_numbers

RACAH does all arithmetic with a subset of the complex numbers, known as c_numbers, that is closed under addition and multiplication. A c_number is a PASCAL record or sequence of records, each containing a numerator, a denominator and a surd, as well as fields that specify its sign and whether it is imaginary. A complex number such as $a + ib$ is a two element linked list, one part of which contains $a$ and the other $b$. The routines to handle these linked lists had been written by G.R.Black and P.H.Butler using integers the size of a machine word. I further modified these routines to make use of the infinite precision integer routines I had implemented.

The program maintains three linked lists for handling c_numbers. All values which are to be preserved, such as the value of a $6j$, are stored in one list. All temporary values that occur in a calculation are on the temporary list, and temporary values that are no longer needed are on the free list. Procedures are available to mark and cut the temporary list so that all intermediate steps that are no longer required can be discarded onto the free list. Memory is only requested from the system when the free list is empty, so available memory is efficiently reused.

Functions are provided that can input, output, add and multiply c_numbers. To subtract c_numbers it is necessary to add the negative by using c_negate and then add. Similarly it is necessary to invert a c_number with c_invert and then multiply if you wish to divide. The program can also conjugate a c_number or convert an integer into c_number format. Due to the format of a c_number it is only possible to take the square root if it does not already contain a surd.

## 8.2 Long Integers

For most groups (such as $SU_3$) it was quite adequate to store the dimensions of irreps as the 32 bit integers that are available on most modern CPU's (16 bit integers as on IBM PC's are not adequate since the intermediate steps of a calculation will overflow these even in $SU_3$, although most compilers for these have a software 32 bit option). When we started to consider the larger Lie groups as possible areas of calculation it became obvious that a 32 bit integer would not be adequate, since the square of the dimension of even the low power irreps of $E_8$ will rapidly overflow the maximum value of $2^{31} - 1$ or 2 147 483 647. To avoid this we decided to implement routines to handle integers to any precision, the basic outline of these routines coming from Knuth(1973). It did not require any major alteration to the c_number routines to achieve this since they are essentially independent of the form in which the integer is stored. All that was required was to add routines to correctly discard any temporary long integer that was no longer needed.

The long integer routines were implemented as linked lists rather than as records of a finite size, so the amount of storage used depends only on the size of the value being stored, and there is no limit to the size of the integer that can be represented, as would be the case for a fixed length record. The addition and subtraction routines were fairly easily implemented, although these and the other routines are complicated by the use of two global variables l_zero and l_one for simplicity of comparisions, since any value of zero or one must be assigned to these. The multiplication routine was also easy to implement, although it is somewhat slower than the routines for a fixed length value since certain optimisations are then possible.

The division routine is the largest and slowest as usually happens even in hardware due to its complexity. This routine first makes sure the divisor is non-zero and then that it is smaller than what it is dividing, otherwise it stops and returns a quotient of zero. It then uses one of two options, since division can be short cut if the divisor has only one digit. The remainder routine, as currently implemented calls the division routine and then calculates the difference between the divisor times the quotient and the original value.

A comparison routine is also implemented by using the subtraction routine. Finally since they are implemented as linked lists there are the routines l_new and l_dump which create and dispose of long integers. A disposed long integer is not returned to the system but is kept on the free list for reuse.

All of the long integer routines are implemented using a global constant called `base`. This determines the range of values stored in a long integer digit, which is primarily determined by the space required to fit $\texttt{base}^2$ into a machine word. So for a 16 bit machine `base`=100, and for a 32 bit machine `base`=10000. If it is possible to have a 32 bit or 64 bit product on 16 bit or 32 bit machines respectively then the values of base can be 10000 and 100 000 000 respectively. All these values are chosen to the nearest power of ten since this is the easiest way to interface them to the routines which write out a decimal integer. The c_number routines are essentially independent of how the long integer routines work so it is very easy to replace the internal workings of these routines without affecting how the c_numbers work.

## 8.3 Matrices

Since most of the $6j$ and $3jm$ routines solve for the unknowns by producing simultaneous equations in the form $Ax = b$ we needed to be able to store these equations as the matrices that they are. This originally resulted in the definition of a c_matrix as a two dimensional array of c_number and a c_vector as a one dimensional array. This was unsatisfactory since arrays have a fixed length whereas there is no real limit to the size that may be required by the $6j$ routines. It was therefore decided to implement the matrices as a doubly linked list. The matrix is represented by a pointer to the (1,1) element which contains a pointer to the rest of the first row and a pointer to the next row. Similarly for the (n,1)th element. The links between rows for the elements in any column other than the first are not used as they are not really necessary. This also has the advantage of not requiring the definition of a c_vector since it can be treated as an $n \times 1$ matrix for our purposes.

There are five routines for manipulating these matrices. The obvious two for creating and disposing of a matrix, and a third which prints out the simultaneous equations they represent. The fourth routine is used to solve the matrix, and recursively solves an upper triangular matrix (recursively since it must start at the bottom row, whilst the pointers link it from the top down). The last routine is used to take a new row for the matrix, row reduce it, and put it into the matrix if it is independent of the other rows. It is this row reduction routine that the algorithm of chapter 4 uses to automatically ignore an equation if it is linearly dependent on a previous equation.

## 8.4 $6j$ and $3jm$ storage

Any number of $6j$ or $3jm$ may be created by the user in the course of a calculation. For this reason a dynamic storage structure is used. No disposal routines are required since once a symbol has been calculated it will be preserved while the program is running, with the symbol being looked up rather than recalculated if again required. Since we will want relatively fast look up of a $6j$ no matter how large the table is, the symbols are stored in a balanced binary tree, with the tree being rebalanced at each insertion to maintain the optimal look up time. The algorithm comes from Wirth(1976), via sample UCSD PASCAL routines provided for an Apple IIe.

## 8.5 Other data elements

As can be seen from the following file, which lists the constants and types used by RACAH, all other structures in the program are implemented as linked lists. When I began on this project, all groups and branches were stored as arrays, but are now dynamically allocated as they are required. The storage of any irrep, triad or $2jm$ data that the program must load has similarly evolved from arrays to dynamically allocated lists. The branching schemes, reduced matrix elements (rmes) and tranformation coefficients were implemented directly as lists.

RACAH file of constants and types

```
                                                   {File: constype}
const
    length_word=16; blank='                ';
    log_title='PRN         ';
    { line_title ='PRN:           '; not used }
    input_title  ='input          '; {not used, not opened}
    output_title ='output         '; {not used, not opened}
    prefix_6title='f6/            ';
    prefix_3title='f3/            ';
    suffix_title ='.dat           ';
    log_head='SUN 3 Racah     ';

    auto_logfile=0; { 0=logfile only when asked,
                      1=prompt for logfile at start
                      2=logfile every run }
    length_term_line=79; { external size of "output" }
    length_printer_line=80;
    disk_text_limit=0; {If >0 then data beyond it is ignored}

    prompt='>'; continuation_char='%'; { used for file "input" only }
    system_backspace=0; { if 0, auto selection of ascii or ebcdic bs}
    length_in_buff=140; length_out_buff=500;  { internal sizes }
```

```
    margin_logfile=6;

    c_max_rank=100; {for simultaneous eqns, rowreduce and solvematrix}
    max_primes=31; {used for put_c_number}
    c_maxheapsize=0; {Lets you know how large things are growing}

    maxmult=10; ctsrep=70; maxprodrep=150; { < sqrt(maxint) }

    spin_char='s'; tilde_char='~'; sym_label='+-&~            ';
    sqrt_in='#'; sqrt_out='#';
    defaultlabelwidth=3;
    parent_group='parent_group    ';

    scalarrep=1; primrep=2; arbrep=0;

    base=10000; { see io_arith for comments on value for various machines }

type
    l_integer=^l_int_rec;
    l_int_rec=record
      digit:integer;
      next:l_integer;
    end;
    word_type=packed array [1..length_word] of char;
    item_type=(is_word,is_number,is_delimiter,end_of_statement,
               end_of_file);
              { returned by get_item }
    in_item_ask_type=(ask_free,ask_word,ask_string,ask_char);
    file_type=(p_none,p_input,disk_one,disk_two,
               p_output,p_logfile,p_line);
    type_in_record=record
      in_buff:  packed array [1..length_in_buff] of char;
      in_start: boolean;
      in_title: word_type;
      in_echo:  boolean;
      in_index: integer;
      in_last:  integer;
      in_item:  item_type;
      in_word:  word_type;
      in_char:  char;
      in_number:integer;
    end;

    c_op_type=(plus,stop,endrow,endcol);
    c_heapname=(no_heap,temp,safe);
    c_number=^c_record;
    c_record=record
      next: c_number;
      op: c_op_type;
      negative,imaginary:boolean;
      num,den,surd: l_integer;
    end;
```

```
c_rank=1..c_max_rank;
c_index=0..c_max_rank;

c_matrix=^matrixelement;
matrixelement=record
  nextcolumn,nextrow:c_matrix;
  element:c_number;
end;

foundtype=(exist,outside,vanish);
bokselect=(bra,operator,ket);

colindextype=1..3;
sixjindextype=1..6;
permtype=array [0..3] of 0..3; { perm[0] is the sym of perm }
coltype=array [colindextype] of integer;

irreptype=integer;           {1 is the scalar irrep }
multtype=0..maxmult;         {1 is the first term   }
symtype=0..3;                {+ or - etc see sym_label }
group_storage_type=(new_group,product,as_tables,so3
                   ,octahedral,tetrahedral,dihedral,cyclic);
dualtype=(selfdual,parity,tildedual,otherdual);
embedtype=(top_of_scheme,uses_tables,coupled,gi_gtilde,so3so2
          ,so3d00,dn_dm,dn_dmtilde,dn_cn,dn_c2,cn_cm);

triadtype= record
  ir3:  array [colindextype] of irreptype;
  mult3: multtype;
  sym3: symtype;
  sumdimen:c_number;  {used by step3j only}  end;
sixjtype= record
  ir6:  array [sixjindextype] of irreptype;
  mult6:  array [1..4] of multtype;
  sym:array [1..4] of symtype; {sym for [21]}
  value6: c_number; end;
ninejtype= record
  ir9:  array [1..9] of irreptype;
  mult9:  array [1..6] of multtype;
  symcol,symrow: symtype;
  value9: c_number; end;
twojmtype= record
  gprep,subrep:irreptype;
  brmult: multtype;
  brsym:symtype;
  sumdimen:c_number; {used by step2jm only} end;
threejmtype= record
  gpir,subir:  array [colindextype] of irreptype;
  gpmult,submult: multtype;
  mult3jm:  array [colindextype] of multtype;
  sym3jm,gpsym,subsym: symtype; { gpsym,subsym for [21] only }
  star3jm: boolean;
```

```
  value3jm: c_number; end;

grouptype     =^grouprecord;
branchtype    =^branchrecord;
schemetype    =^schemerecord;
gptabletype   =^gptablerecord;
brtabletype   =^brtablerecord;
irreplist     =^irreprecord;
triadlist     =^triadrecord;
twojmlist     =^twojmrecord;
ptr6jnode     =^sixjnode;
ptr3jmnode    =^threejmnode;
boktype       =^bokrecord;
bokheadtype   =^bokheadrecord;
rmetype       =^rmerecord;
rmeheadtype   =^rmeheadrecord;
transtype     =^transrecord;
transheadtype =^transheadrecord;
transtabletype=^transtablerecord;

grouprecord= record
  fullname,barename,nametag:word_type;
  howstored: group_storage_type;
  labelwidth:0..length_word;
  duality:dualtype;
  continuous: boolean;
  nrep: irreptype;
  ntrmult: multtype;
  sixjlocation: grouptype;   { First of isomorphic copies}
  n6j:integer;               {Only kept for debug purposes}
  sixjs:ptr6jnode;
  group1,group2: grouptype; { only used if product}
  gptablelocation: gptabletype; { only used if tables}
  myscheme:schemetype;
  grouplevel,sublevel:branchtype; {Where used in scheme}
  irfixed: boolean; {These two for the bok currently being created}
  irselect: irreptype;
  gprmemult: multtype;
  nextgroup:grouptype; end;
branchrecord= record
  group,subgroup: grouptype;
  howembed: embedtype;
  nbrmult: multtype;
  threejmlocation: branchtype;   { First of isomorphic copies }
  n3jm:integer;
  threejms:ptr3jmnode;
  tilde_rep: irreptype; { for reflection embeddings }
  brtablelocation: brtabletype;
  myscheme:schemetype;
  nextbranch: branchtype; end;

schemerecord= record
```

```
  schemename:word_type;
  topbranch,bottombranch:branchtype;
  bottomgroup:grouptype;  { A product of all non-abelian groups in the
                            scheme that are not branched from }
  firstbokhead,lastbokhead: array[bokselect] of bokheadtype;
  firstrmehead,lastrmehead:rmeheadtype;
  nextscheme:schemetype; end;
gptablerecord= record
  nameinfile,filetitle,filetime,filedate: word_type;
  irrepdata:irreplist;
  ntriads:  integer;
  triaddata:triadlist; end;
brtablerecord= record
  filetitle,nameinfile,filetime,filedate: word_type;
  n2jm : integer;
  lastgprep_intable: irreptype;
  twojmdata: twojmlist; end;
irreprecord= record
  irname: word_type;
  conj,power:irreptype;
  dimen: c_number;
  twoj: symtype;
  nextirrep: irreplist; end;
triadrecord= record
  triad: triadtype;
  nexttriad: triadlist; end;
twojmrecord=record
  twojm: twojmtype;
  nexttwojm:twojmlist; end;
sixjnode =  record
  item:   sixjtype;
  left,right: ptr6jnode;
  bal: -1..1; end;
threejmnode= record
  item: threejmtype;
  left,right: ptr3jmnode;
  bal: -1..1; end;


bokheadrecord=  record
  bokname :word_type;
  myboks  :boktype;
  firstbotbok,
  lastbotbok :boktype;   { Allows access to bottom level }
  howmanyboks :integer;  { Used in creating mynumber }
  nextbokhead:bokheadtype; end;
bokrecord     =  record
  { the top irrep is the subir of the first bokrecord -
    the bottom irrep has to be collected together and is bottomir }
  gpir,subir: irreptype;
  mult       :multtype;     { sub-branching mult - complicated! }
  myhead     :bokheadtype;  { only used at bottom }
  mynumber   :integer;      { ''. "Multiplicity - <= howmanyboks. }
```

```
    bottomir  :irreptype;     { ''}
    mydimension:c_number;     { '' - used? }
    daughter,
    sister,
    cousin    : boktype; end;  { '' connects all with same head }
  rmerecord= record
    rmeboks  :array [bokselect] of boktype;
    rmebottommult :multtype;
    rmevalue :c_number;
    myrmehead :rmeheadtype;
    nextrme:rmetype; end;
  rmeheadrecord=record
    rmebokheads :array[bokselect] of bokheadtype;
    rmetopmult  :multtype;
    reducedme   :c_number;
    firstrme,
    lastrme     :rmetype;
    nextrmehead :rmeheadtype;
  end;
  transrecord=record
    leftstate, rightstate :boktype;
    transvalue :c_number;
    nexttrans :transtype;
  end;
  transheadrecord=record
    leftbokhead, rightbokhead :bokheadtype;
    firsttrans, lasttrans :transtype;
    nexttranshead :transheadtype;
  end;
  transtablerecord=record
    leftscheme, rightscheme :schemetype;
    firsttranshead, lasttranshead :transheadtype;
    nexttranstable :transtabletype;
  end;
```

## 8.6 Input

The first element of any input is required to be a command word. The various commands are contained in three command interpreters. All c_number commands and file handling are interpreted by get_command, all irrep, $3j$, $6j$, $3jm$, group and branch commands are in racah_command and all bra, ket, scheme, transformation factor and reduced matrix element (rme) commands are interpreted by wigner_command. Each of these three procedures is called before a command is rejected as unknown. The three command interpreters are listed below.

```
function get_command{:boolean};
  {Loops until either not one of its commands (returns false) or
   all files closed  (returns true). }
var understood:boolean;
    marker:c_number;
begin
  with in_record[in_file] do begin
    understood:=true;
    repeat
      if (in_item<>end_of_statement) and (in_item<>end_of_file) then
        get_item(ask_word);
      if (in_item<>end_of_statement)and(in_item<>end_of_file) then begin
        put_string('end of statement'); put_string(' expected,;     ');
        put_string(' input skipped: '); put_input; send_warn;
        repeat get_item(ask_word);
        until (in_item=end_of_statement) or (in_item=end_of_file);
      end;
      while in_item=end_of_statement do get_item(ask_word);
      if in_item=is_delimiter then begin put_string('command word exp');
        put_string('ected;          '); put_input; send_warn; end else
      if in_item=is_word then begin
        if in_word='logfile         ' then begin
          get_item(ask_string);
          if in_item<>is_word then in_word:=log_title;
          open_file(p_logfile,in_word,log_head) end else
        if in_word='nologfile       ' then logging:=false else
        if in_word='debug           ' then debug:=true else
        if in_word='nodebug         ' then debug:=false else
        if in_word='input           ' then begin
          get_item(ask_string);
          if in_item=is_word then begin
            open_file(in_file,in_word,blank);
            in_record[in_file].in_echo:=true;
            understood:=get_command; end
          else begin put_string('File title neede'); put_char('d');
            put_input; send_warn; end; end else
        if in_word='finish          ' then in_item:=end_of_file else
        if in_word='c_number         ' then begin
          c_markheap(marker);
          put_c_number(get_c_number);send;
          c_cutheap(marker); end else
        understood:=false; end;
    until (not understood) or (in_item=end_of_file);
    if in_item=end_of_file then begin
      put_string('File ;          ');
      put_name(in_title); put_string(' closed;        '); send;
      close_file(in_file); end; end;
  get_command:=understood;
end;

function racah_command:boolean;
var c_here:c_number;
```

```
begin racah_command:=true; c_markheap(c_here);
  with in_record[in_file] do
  if      in_word='spinlabel       ' then quotespin:=false
  else if in_word='quotelabel      ' then quotespin:=true
  else if in_word='indexin         ' then indexin:=true
  else if in_word='labelin         ' then indexin:=false
  else if in_word='indexout        ' then indexout:=true
  else if in_word='labelout        ' then indexout:=false
  else if in_word='group           ' then setgroup
  else if in_word='branch          ' then setbranch
  else if in_word='groups          ' then showgpdata(nilgroup)
  else if in_word='branches        ' then showbrdata(nilbranch)

  else if cur_group=nilgroup then no_data('groups          ')
  else if in_word='labelwidth      ' then setlabelwidth
  else if in_word='ir              ' then info_ir
  else if in_word='irs             ' then tableirreps
  else if in_word='showir          ' then showirrep
  else if in_word='3j              ' then info_3j
  else if in_word='3js             ' then table3js
  else if in_word='6j              ' then info_6j
  else if in_word='input6j         ' then input6j
  else if in_word='6js             ' then table6js
  else if in_word='9j              ' then info_9j
  else if in_word='show6js         ' then show6js
  else if in_word='step3j          ' then loop3j
  else if in_word='step6j          ' then loop6j

  else if cur_branch=nilbranch then no_data('branches        ')
  else if in_word='2jm             ' then info_2jm
  else if in_word='2jms            ' then table2jm
  else if in_word='3jm             ' then info_3jm
  else if in_word='show3jms        ' then show3jms
  else if in_word='input3jm        ' then input3jm
  else if in_word='3jms            ' then table3jms
  else if in_word='step2jm         ' then loop2jm
  else if in_word='step3jmgroup    ' then loop3jmgroup
  else if in_word='step3jmsub      ' then loop3jmsub
  else racah_command:=false;
  c_cutheap(c_here);
  end;

  function wigner_command{:boolean};
  begin wigner_command:=true;
    with in_record[in_file] do
    if      in_word='scheme          ' then
                             cur_scheme:=getscheme(cur_scheme)
    else if cur_scheme=nilscheme then wigner_command:=false
    else if in_word='schemes         ' then showscdata(nilscheme)
    else if in_word='ket             ' then getbok(ket,cur_scheme)
    else if in_word='kets            ' then tablebok(ket,cur_scheme)
```

```
    else if in_word='bra          ' then getbok(bra,cur_scheme)
    else if in_word='bras         ' then tablebok(bra,cur_scheme)
    else if in_word='op           ' then getbok(operator,cur_scheme)
    else if in_word='ops          ' then tablebok(operator,cur_scheme)
    else if in_word='rme          ' then getrme(cur_scheme)
    else if in_word='rmes         ' then showrmes(cur_scheme,false)
    else if in_word='prettyrmes   ' then showrmes(cur_scheme,true)
    else if in_word='transtable   ' then
                           cur_transtable:=gettranstable(cur_transtable)
    else if cur_transtable=niltranstable then wigner_command:=false
    else if in_word='transtables  ' then showtranstable(niltranstable)
    else if in_word='storetrans   ' then storetrans(cur_transtable)
    else if in_word='showtrans    ' then
                           showtrans(cur_transtable,arbrep,false)
    else if in_word='prettytrans  ' then prettytrans(cur_transtable)
    else if in_word='dotrans      ' then dotrans(cur_transtable)
    else wigner_command:=false;
  end;
```

## 8.7 Selection Rule Data

The information required to generate the products and branchings of various point groups and $SO_3$ is contained within RACAH. The user must provide the irrep,product and branching data for RACAH to load if the group is unknown to the program (e.g. most Lie groups). To calculate $6j$ it is only necessary to provide data about the group, while $3jm$, kets etc require a subgroup chain and the branching rules. This information is described in files of the format described below.

The file of group data must contain irrep and triad information, and may contain $6j$ data. The first line of the file is the heading and the presence or absence of the word `CONTINUOUS` at the end of the line tells RACAH whether the group is continuous or finite. The first data section is started with the word `irreps`. Each field is separated by a comma, with the end of the section being signaled by a semi-colon. The irrep field contains the irrep's label, the $2j$ phase and its dimension. The label of the conjugate irrep may also be present. The triad fields are separated in the same way as the irreps and each field contains the three irrep labels, the multiplicity label and the symmetry type of the triad (generated from the symmetrised power). If the $6j$ data is present then the irrep and multiplicity labels used are the programs internal labels, not the irrep labels. Data tables are shown below for $G_2$ and $SU_3$. The $G_2$ file contains $6j$ data and the $SU_3$ file is an example of a group with complex irreps. The default names for these files are generated by concatenating the `prefix_6title`, the group name and the `suffix_title` (these objects are in the file of constants listed above) and are `f6/g2.dat` and `f6/su3.dat` respectively.

## $G_2$ data file

```
GROUPDATA G2 created Thursday 30_June_1988 CONTINUOUS
irreps  0  +1,  1  +7,  2  +27, 21  +14,  3  +77, 31  +64;
triads 0 0 0 0+, 1 1 0 0+, 1 1 1 0-, 2 1 1 0+, 2 2 0 0+, 2 2 1 0-,
2 2 2 0+, 2 2 2 1+, 21 1 1 0-, 21 2 1 0+, 21 2 2 0-, 21 21 0 0+,
21 21 2 0+, 21 21 21 0-, 3 2 1 0+, 3 2 2 0-, 3 21 2 0+, 3 21 21 0-,
3 3 0 0+, 3 3 1 0-, 3 3 2 0+, 3 3 2 1+, 3 3 21 0-, 3 3 3 0-, 3 3 3 1-,
31 2 1 0-, 31 2 2 0+, 31 2 2 1-, 31 21 1 0+, 31 21 2 0-, 31 3 1 0+,
31 3 2 0-, 31 3 2 1+, 31 3 21 0+, 31 3 3 0+, 31 3 3 1-, 31 31 0 0+,
31 31 1 0-, 31 31 2 0+, 31 31 2 1+, 31 31 21 0-, 31 31 21 1-,
31 31 3 0-, 31 31 3 1+, 31 31 3 2-, 31 31 31 0+, 31 31 31 1-;
6jtable       2   2   2    2   2   2    1 1 1 1 - 1/2.7,
      2   2   3    2   2   2    1 1 1 1 + 1/2.3.7,
      2   2   3    2   2   3    1 1 1 1 + 5/2.9.3.7,
      2   2   4    2   2   2    1 1 1 1 + 1/2.7,
      2   2   4    2   2   3    1 1 1 1 + 1/2.3.7,
      2   2   4    2   2   4    1 1 1 1 + 0;
```

## $SU_3$ data file

```
GROUPDATA SU3 created for_Searle(1986) 26_August_1986 CONTINUOUS
irreps  0  +1,  1  * 11  +3,  2  * 22  +6,  21 +8,  3  * 33  +10,
31 *  32  +15;
triads 0 0 0 0+, 1 1 1 0-, 11 1 0 0+, 2 11 11 0+, 2 2 2 0+, 22 2 0 0+,
21 11 1 0+, 21 2 1 0-, 21 22 2 0+, 21 21 0 0+, 21 21 21 0+, 21 21 21 1-,
3 22 11 0+, 3 21 21 0-, 3 3 3 0-, 33 3 0 0+, 33 3 21 0+, 31 22 1 0+,
31 22 22 0-, 31 21 11 0+, 31 21 2 0-, 31 3 2 0+, 31 33 11 0-,
31 33 2 0+, 31 31 1 0-, 31 31 22 0+, 31 31 31 0-, 31 31 31 1+,
32 31 0 0+, 32 31 21 0+, 32 31 21 1-, 32 31 3 0-;
```

The branching files must contain branching data and may contain $3jm$ data. The branching data lists the group irrep followed by the subgroup irreps to which it branches and the sign of the associated $2jm$. As for the group data the $3jm$ data uses the internal program labels for the irreps and the mutilpicity. The file for branching $G_2$ to $SO_3$ is listed below and has the default name `f3/g2so3.dat`.

## $G_2$ to $SO_3$

```
branchdata G2_to_SO3 created Thursday 30_Jun_1988
branchings
0    +0 ,
1    +3 ,
2    +2 + 4 +6 ,
21   +1 + 5 ,
3    + 1 + 3 + 4 + 5 + 6 + 7 + 9 ,
31   + 2 + 3 + 4 + 5 + 7 + 8;
```

# 8.8 SCHUR and plethysms

To be able to attempt any calculation of $6j$ for the group $E_8$ it was necessary to be able to generate the product rules and symmetrised products for the group. The products for the group are readily calculated by the program SCHUR (written by G.R.Black, B.G.Wybourne and others) and are readily converted into a data table of the format required by RACAH. The symmetrised powers were a more difficult problem.

To calculate the plethysms of $E_8$ the irrep was first branched down to $SU_9$, where the symmetrised power is just the Schur function plethysm, with the resulting partitions modified appropriately for $SU_9$. The algorithms used were based upon M. Wilson's algorithm which later became part of SCHUR's commands. The plethysm routines were modified to truncate the partitions as they were generated rather than at the end of the calculation. This limited the amount of storage required since some calculations were close to using all available memory. This also speeded up the calculation slightly since only the partitions that would contribute to the result were actually involved in the intermediate steps. The routines were also extended to allow the calculation of plethysms involving products of irreps, rather than just sums of irreps, so that if a problem could be written simply as a product the calculation could also be simplified and speeded up.

Once the plethysm had been calculated it was necessary to calculate $E_8$ to $SU_9$ branching rules larger than those available (up to $(94^63)$). This was accomplished by taking $E_8$ products that only produced one unknown irrep, then taking the $SU_9$ product of the branched irreps. A routine was written to automate the process of subtracting out the known branching rules to find the branching rule of the unknown. These results were then checked by making sure that they were dimensionally correct, and that the Dynkin indices were consistent, and that the result was real.

Once this had been done it was then possible to use another routine to restate the $SU_9$ plethysm in terms of $E_8$ irreps, since the highest weight of the branching is unique. This required that no $SU_9$ irreps were left over once the process was complete, and the plethysm had to be real. A final check applied to the symmetrised powers was that the sum of the various squares equalled the square of the irrep. This made it possible to resolve up to the cubes of $(21)$, $(31^6)$ and $(42^7)$ and the squares of $(3)$,$(41^5)$,$(421^6)$, $(52^61)$ and $(63^7)$, whereas only the powers of the primitive, $(21^7)$, and one power two irrep, $(21)$, had been available (see Wybourne 1979 and McKay et al 1981). The results are listed below, with the $E_8$ irreps being labelled by the highest $SU_9$ irrep in the branching.

Table 8.1: $E_8$ Plethysms

$$(21^7)\otimes\{2\} = (42^7)+(21)+(0)$$

$$(21^7)\otimes\{1^2\} = (31^6)+(21^7)$$

$$(21^7)\otimes\{3\} = (63^7)+(421^6)+(31^6)+(21^7)$$

$$(21^7)\otimes\{21\} = (52^61)+(42^7)+(421^6)+(31^6)+(3)+2(21^7)+(21)$$

$$(21^7)\otimes\{1^3\} = (42^7)+(41^5)+(31^6)+(21)+(0)$$

$$(21)\otimes\{2\} = (42^7)+(42)+(41^5)+(3)+(21)+(0)$$

$$(21)\otimes\{1^2\} = (421^6)+(41^2)+(31^6)+(21^7)$$

$$\begin{aligned}(21)\otimes\{3\} = {} & (632^6)+(63)+(62^6)+(62^21^5)+(621^4)+(52^61)\\ & +(521^5)+2(51^7)+(51^4)+(51)+2(42^7)+2(421^6)\\ & +2(42)+3(41^5)+(41^2)+(31^6)+(3)+3(21)+(0)\end{aligned}$$

$$\begin{aligned}(21)\otimes\{21\} = {} & 2(632^6)+(631^6)+(62^51^1)+(62^21^5)+(621^4)+(621)\\ & +(61^6)+3(52^61)+3(521^5)+3(51^7)+2(51^4)+2(51)\\ & +2(42^7)+5(421^6)+2(42)+3(41^5)+3(41^2)+3(31^6)\\ & +4(3)+2(21^7)+3(21)\end{aligned}$$

$$\begin{aligned}(21)\otimes\{1^3\} = {} & (63^7)+(631^6)+(62^51^1)+(62^21^5)+(61^3)+(52^61)\\ & +2(521^5)+2(51^7)+(51)+3(421^6)+(41^5)+2(41^2)\\ & +3(31^6)+(3)+(21^7)+(21)\end{aligned}$$

$$\begin{aligned}(31^6)\otimes\{2\} = {} & (632^6)+(62^6)+(52^61)+(51^7)+(51^4)+2(42^7)\\ & +(421^6)+(42)+2(41^5)+(3)+2(21)+(0)\end{aligned}$$

$$\begin{aligned}(31^6)\otimes\{1^2\} = {} & (63^7)+(62^51^1)+(52^61)+(521^5)+(51^7)+2(421^6)\\ & +(41^2)+2(31^6)+(3)+(21^7)\end{aligned}$$

$$\begin{aligned}(31^6)\otimes\{3\} = {} & (94^63)+(943^52)+(93^6)+2(843^6)+(83^61)+2(83^52^2)\\ & +2(83^22^5)+(832^41^2)+2(82^61)+(82^41^2)+7(73^62)\\ & +7(732^51)+2(731^5)+5(72^7)+4(72^51)+4(72^41^3)\\ & +(72^21^4)+3(721^6)+(71^5)+(71^2)+5(63^7)+8(632^6)\\ & +5(631^6)+3(62^6)+12(62^51^1)+9(62^21^5)+5(621^4)\\ & +(621)+8(61^6)+5(61^3)+11(52^61)+16(521^5)\\ & +14(51^7)+5(51^4)+7(51)+3(42^7)+14(421^6)+2(42)\\ & +9(41^5)+11(41^2)+11(31^6)+5(3)+3(21^7)+3(21)\end{aligned}$$

8.1 continued

$$
\begin{aligned}
(31^6)\otimes\{21\} \;=\;& (94^63)+(943^52)+(93^7)+(93^521)+2(84^7)+4(843^6)\\
&+(842^6)+3(83^61)+5(83^52^2)+3(83^22^5)+(832^5)\\
&+2(832^41^2)+3(82^61)+(82^41^2)+(82^31^4)+11(73^62)\\
&+14(732^51)+2(731^5)+11(72^7)+6(72^51)+8(72^41^3)\\
&+4(72^21^4)+6(721^6)+(72111)+3(71^5)+6(63^7)\\
&+20(632^6)+9(631^6)+9(62^6)+22(62^51^1)+18(62^21^5)\\
&+11(621^4)+4(621)+16(61^6)+5(61^3)+2(6)\\
&+25(52^61)+28(521^5)+26(51^7)+16(51^4)+14(51)\\
&+10(42^7)+26(421^6)+9(42)+20(41^5)+18(41^2)\\
&+15(31^6)+15(3)+6(21^7)+8(21)
\end{aligned}
$$

$$
\begin{aligned}
(31^6)\otimes\{1^3\} \;=\;& (94^63)+(93^42^3)+(84^7)+2(843^6)+(842^6)+(83^61)\\
&+3(83^52^2)+(83^22^5)+(832^5)+(832^41^2)+2(82^61)\\
&+(82^5)+4(73^62)+7(732^51)+6(72^7)+2(72^51)\\
&+4(72^41^3)+3(72^21^4)+3(721^6)+(721^3)+(71^5)\\
&+4(63^7)+11(632^6)+4(631^6)+(63)+8(62^6)\\
&+10(62^51^1)+10(62^21^5)+7(621^4)+(621)+6(61^6)\\
&+2(61^3)+2(6)+11(52^61)+12(521^5)+14(51^7)\\
&+9(51^4)+7(51)+7(42^7)+12(421^6)+7(42)\\
&+13(41^5)+7(41^2)+6(31^6)+7(3)+(21^7)\\
&+6(21)+2(0)
\end{aligned}
$$

$$
\begin{aligned}
(42^7)\otimes\{2\} \;=\;& (84^7)+(632^6)+(62^6)+(52^61)+2(42^7)+(42)+(41^5)\\
&+(21)+(0)
\end{aligned}
$$

$$
(42^7)\otimes\{1^2\} \;=\; (73^62)+(63^7)+(52^61)+(521^5)+(421^6)+(31^6)+(21^7)
$$

$$
\begin{aligned}
(42^7)\otimes\{3\} \;=\;& (12\,6^7)+(10\,54^6)+(10\,4^62)+2(94^63)+(943^52)\\
&+3(84^7)+(843^6)+2(842^6)+(83^61)+3(83^52^2)+(832^5)\\
&+(82^61)+(82^5)+4(73^62)+4(732^51)+2(72^7)+(72^51)\\
&+(72^41^3)+(72^21^4)+(721^6)+2(63^7)+7(632^6)+(631^6)\\
&+(63)+6(62^6)+2(62^51^1)+3(62^21^5)+3(621^4)+2(61^6)\\
&+(6)+5(52^61)+4(521^5)+4(51^7)+2(51^4)+2(51)\\
&+6(42^7)+3(421^6)+4(42)+6(41^5)+(41^2)+2(31^6)\\
&+(3)+3(21)+2(0)
\end{aligned}
$$

8.1 continued

$$\begin{aligned}
(42^7) \otimes \{21\} = & (11\ 5^64) + (10\ 5^7) + (10\ 54^6) + (10\ 4^62) + 3(94^63) \\
& + 2(943^52) + (93^7) + (93^521) + 4(84^7) + 4(843^6) \\
& + 2(842^6) + 4(83^61) + 4(83^52^2) + (83^22^5) + (832^5) \\
& + (832^41^2) + (82^61) + 9(73^62) + 8(732^51) + (731^5) \\
& + 4(72^7) + 3(72^51) + 2(72^41^3) + (72^21^4) + 2(721^6) \\
& + (71^5) + 5(63^7) + 10(632^6) + 4(631^6) + 6(62^6) \\
& + 8(62^51^1) + 5(62^21^5) + 4(621^4) + (621) + 5(61^6) \\
& + (6) + 13(52^61) + 10(521^5) + 7(51^7) + 4(51^4) + 4(51) \\
& + 7(42^7) + 9(421^6) + 4(42) + 6(41^5) + 4(41^2) + 5(31^6) \\
& + 4(3) + 3(21^7) + 3(21)
\end{aligned}$$

$$\begin{aligned}
(42^7) \otimes \{1^3\} = & (10\ 5^7) + (10\ 4^53^2) + 2(94^63) + (943^52) + (93^6) \\
& + 3(843^6) + 2(83^61) + (83^52^2) + (83^22^5) + (832^41^2) \\
& + (82^61) + 5(73^62) + 4(732^51) + (731^5) + 2(72^7) \\
& + 2(72^51) + (72^41^3) + (721^6) + 5(63^7) + 3(632^6) \\
& + 3(631^6) + 2(62^6) + 6(62^51^1) + 2(62^21^5) + (621^4) \\
& + 2(61^6) + 2(61^3) + 5(52^61) + 6(521^5) + 4(51^7) + (51^4) \\
& + 2(51) + (42^7) + 6(421^6) + 2(41^5) + 3(41^2) \\
& + 5(31^6) + (3) + 2(21^7)
\end{aligned}$$

$$\begin{aligned}
(3) \otimes \{2\} = & (632^6) + (62^6) + (62^21^5) + (621^4) + (6) + (52^61) \\
& + (521^5) + (51^7) + (51^4) + (51) + 2(42^7) \\
& + (421^6) + 2(42) + 2(41^5) + (3) + (21) + (0)
\end{aligned}$$

$$\begin{aligned}
(3) \otimes \{1^2\} = & (63^7) + (631^6) + (62^51^1) + (61^6) + (61^3) \\
& + (52^61) + 2(521^5) + (51^7) + (51) + 2(421^6) \\
& + 2(41^2) + 2(31^6) + (21^7)
\end{aligned}$$

$$\begin{aligned}
(421^6) \otimes \{2\} = & (84^7) + (842^6) + (83^52^2) + (73^62) + 2(732^51) + 2(72^7) \\
& + (72^41^3) + (72^21^4) + (721^6) + 5(632^6) + (631^6) \\
& + (63) + 3(62^6) + 2(62^51^1) + 4(62^21^5) + 3(621^4) + (621) \\
& + 2(61^6) + (6) + 4(52^61) + 4(521^5) + 5(51^7) + 4(51^4) \\
& + 3(51) + 4(42^7) + 3(421^6) + 4(42) + 6(41^5) + 2(41^2) \\
& + (31^6) + 3(3) + 3(21) + (0)
\end{aligned}$$

8.1 continued

$$(421^6) \otimes \{1^2\} = (843^6) + (83^22^5) + 2(73^62) + 2(732^51) + (731^5) + (72^7) + (72^51) + (72^41^3) + (721^6) + 2(63^7) + 2(632^6) + 3(631^6) + 5(62^51^1) + 3(62^21^5) + (621^4) + (621) + 3(61^6) + 2(61^3) + 4(52^61) + 6(521^5) + 5(51^7) + 2(51^4) + 3(51) + 6(421^6) + 2(41^5) + 5(41^2) + 4(31^6) + 2(3) + 2(21^7)$$

$$(41^5) \otimes \{2\} = (84^7) + (842^6) + 2(83^52^2) + (832^5) + (82^61) + (82^5) + (82^31^4) + (73^62) + 3(732^51) + 2(72^7) + (72^51) + 2(72^41^3) + 2(72^21^4) + 2(721^6) + (721^3) + (71^5) + 5(632^6) + (631^6) + (63) + 4(62^6) + 2(62^51^2) + 5(62^21^5) + 4(621^4) + (621) + 3(61^6) + 2(6) + 3(52^61) + 3(521^5) + 4(51^7) + 4(51^4) + 3(51) + 3(42^7) + 2(421^6) + 4(42) + 5(41^5) + (41^2) + 2(3) + 2(21) + (0)$$

$$(41^5) \otimes \{1^2\} = (843^6) + (83^61) + (83^22^5) + (832^41^2) + (82^61) + (82^41^2) + 2(73^62) + 3(732^51) + (731^5) + 2(72^7) + 2(72^51) + 2(72^41^3) + (72^21^4) + 2(721^6) + (71^5) + (71^2) + 2(63^7) + 2(632^6) + 3(631^6) + 5(62^51^1) + 3(62^21^5) + 2(621^4) + (621) + 4(61^6) + 3(61^3) + 3(52^61) + 6(521^5) + 4(51^7) + 2(51^4) + 3(51) + 4(421^6) + (41^5) + 4(41^2) + 3(31^6) + (3) + (21^7)$$

$$(52^61) \otimes \{2\} = (10\ 54^6) + (10\ 4^62) + 2(94^63) + (943^52) + (93^7) + (93^521) + (93^42^3) + 3(84^7) + 2(843^6) + 2(842^6) + 2(83^61) + 5(83^52^2) + (83^22^5) + 2(832^5) + (832^41^2) + 2(82^61) + (82^5) + 4(73^62) + 6(732^51) + 4(72^7) + 2(72^51) + 3(72^41^3) + 2(72^21^4) + 2(721^6) + (721^3) + (71^5) + (63^7) + 8(632^6) + 2(631^6) + (63) + 6(62^6) + 4(62^51^1) + 5(62^21^5) + 5(621^4) + (621) + 4(61^6) + (6) + 5(52^61) + 4(521^5) + 5(51^7) + 4(51^4) + 3(51) + 4(42^7) + 3(421^6) + 4(42) + 6(41^5) + 2(41^2) + (31^6) + 2(3) + 3(21) + (0)$$

8.1 continued

$$
\begin{aligned}
(52^61)\otimes\{1^2\} \;=\;& (10\ 5^7)+(10\ 4^53^2)+2(94^63)+2(943^52)+(93^7)\\
&+(93^6)+(93^521)+4(843^6)+3(83^61)+2(83^52^2)\\
&+2(83^22^5)+2(832^41^2)+2(82^61)+(82^41^2)+6(73^62)\\
&+6(732^51)+2(731^5)+4(72^7)+4(72^51)+3(72^41^3)\\
&+(72^21^4)+2(721^6)+(71^5)+4(63^7)+3(632^6)\\
&+4(631^6)+(62^6)+8(62^51^1)+4(62^21^5)+2(621^4)\\
&+(621)+4(61^6)+3(61^3)+5(52^61)+8(521^5)+5(51^7)\\
&+2(51^4)+3(51)+6(421^6)+(41^5)+4(41^2)+4(31^6)\\
&+2(3)+2(21^7)
\end{aligned}
$$

$$
\begin{aligned}
(63^7)\otimes\{2\} \;=\;& (12\ 6^7)+(10\ 54^6)+(10\ 4^62)+(94^63)+2(84^7)\\
&+(842^6)+(83^61)+(83^52^2)+(832^5)+(73^62)\\
&+(732^51)+2(632^6)+(63)+2(62^6)+(621^4)\\
&+(52^61)+2(42^7)+(42)+(41^5)+(21)+(0)
\end{aligned}
$$

$$
\begin{aligned}
(63^7)\otimes\{1^2\} \;=\;& (11\ 5^64)+(10\ 5^7)+(94^63)+(943^52)+(93^6)\\
&+(843^6)+(83^61)+2(73^62)+(732^51)+(731^5)\\
&+(72^51)+2(63^7)+(631^6)+(62^51^1)+(52^61)\\
&+(521^5)+(421^6)+(31^6)+(21^7)
\end{aligned}
$$

# Chapter 9

# Conclusion

The calculation of $6j$ only requires the product rules and symmetrised powers of a group, it is completely independent of the basis choice for the $3jm$. The properties of the $6j$ symbols for an arbitrary compact group have been reviewed. The possible choices for the complex conjugation and permutation matrices have been discussed, and convenient values chosen.

The algorithm of Butler and Wybourne(1976) for calculating all $6j$ and $3jm$ symbols in terms of the primitive symbols has been substantially improved. This algorithm has been shown to apply to a wider category of transformation factors including the induction factors of Haase and Butler(1984).

The set of primitive $6j$ has been split into four classes and complete algorithms specified for the calculation of three of these classes. The fourth class of $6j$, known as the core $6j$, has been shown to contain the coupling phase choices for the Racah-Wigner algebra and a definition of the basis $6j$ has been given. The $6j$ have also been shown to contain orientation phase choices similar to those found by Reid and Butler(1980) for $3jm$.

A complete algorithm for the calculation of core $6j$ has not been specified. A proof of the completeness of the method has been given for the group $SO_3$ and the special cases that occur in more complex groups have been discussed. Values for some $6j$ of $G_2$ and $E_8$ have been produced using this method, and using the results for the mixed symmetry matrices we have completely solved the $6j$ for the finite group $K_{20}$. This algorithm is conjectured to be complete as it has been impossible to find any compact group for which it fails.

The algorithms for the calculation of all non-core $6j$ have been implemented in a PASCAL program RACAH. Algorithms have also been written to allow the symmetrised powers of the classical Lie groups to be calculated with the program SCHUR.

# References


Albrow, M.D., 1988, B.Sc(Hons) Project, University of Canterbury.

Bickerstaff R.P., 1981, PhD thesis, University of Canterbury, New Zealand.

Bickerstaff R.P., Butler P.H., Butts M.B., Haase R.W., and Reid M.F., 1982, J. Phys A **15**, 1087-117

Bickerstaff R.P., 1984, J Math Phys **25**, 2808-14

Bickerstaff R.P. and Damhus T., 1985, Int. J. Quantum Chem. **27**, 381-91

Biedenharn L.C., Brouwer W. and Sharp W.T., 1968, The Algebra of Representations of some Finite Groups, Rice University (Houston)

Bovier A., Lüling M. and Wyler D., 1981, J. Math Phys **22**, 1536-42

Butler P.H., 1975, Phil. Trans R. Soc A **277**, 545-85

Butler P.H., 1976, Int J Quantum Chem **10**, 599-613

Butler P.H., 1981, Point Group Symmetry Applications: Methods and Tables, Plenum (New York)

Butler P.H. and King R.C., 1974, Can J Math **26**, 328-339

Butler P.H. and Wybourne B.G., 1976, Int J Quantum Chem. **10**, 581-98

Chen J-Q., Gao M-J. and Ma G-Q, 1985, Rev Mod Phys **57**, 211-78

Derome J.R., 1966, J. Math Phys **7**, 612-5

Derome J.R. and Sharp W.T., 1965, J. Math Phys **6**, 1584-90

Gao M-J. and Chen J-Q., 1985, J. Phys A **18**, 189-213

Hamer C.J., Irving A.C. and Preece T.E., 1986, Nucl Phys B **270**, 553-74

Haase R.W. and Butler P.H., 1984, J. Phys A **17**, 47-59

Haase R.W. and Butler P.H., 1985, J. Math Phys **26**, 1493-513

Haase R.W. and Dirl R., 1986, J. Math Phys **27**, 900-13

Judd B.R., 1986, J. Math Phys **27**, 2616-22

Judd B.R., 1987, J. Phys A **20**, L343-7

Judd B.R., Lister G.M.S. and O'Brien M.C., 1986, J. Phys A **19**, 2473-86

Knuth D.E., 1973, The Art of Computer Programming Vol 1. 2nd Edition, Addison-Wesley(Reading, Mass.)

McKay J., Patera J. and Sharp R.T., J. Math Phys **22**, 2770-74

Messiah, A., 1961, Quantum Mechanics , North Holland (Amsterdam).

Newmarch J.D., 1983, J. Math Phys **24**, 451-6

Pluhar Z., Weigert L.J. and Holan P., 1986, J. Phys A **19**, 29-34

Racah G., 1949, Phys Rev **76**, 1352-65

Raynal J. and Conté R., 1985, J Math Phys **26**, 2413-40

Reid M.F. and Butler P.H., 1980, J Phys A **13**, 2889-901

Reid M.F. and Butler P.H., 1982, J Phys A **15**, 2327-35

Searle B.G. and Butler P.H., 1988a, J Phys A **21**, 1977-81

Searle B.G. and Butler P.H., 1988b, J Phys A **21**, 3041-50

Searle B.G. and Butler P.H., 1988c, J Phys A **21**, 3313-19

Sullivan J.J., 1983, J. Math Phys **24**, 424-40

Wirth N., 1976, Algorithms+data structures = programs, Prentice-Hall (New Jersey)

Wybourne B.G., 1970, Symmetry principles and atomic spectroscopy, Wiley (New York).

Wybourne B.G., 1979, Aust J. Phys **32**, 417-26

Wybourne B.G. and Bowick M.J., 1977, Aust J. Phys **30**, 259-86

Zhang Q. and Xiangzhu L., 1987, J Phys A **20**, 6185-96

Zeng G.J., 1987, J Phys A **20**, 1961-71

# Appendix A

# Publications

Parts of this thesis have already been jointly published with P.H.Butler. The contents of chapter 4 have been published in

Searle B.G. and Butler P.H., 1988, J Phys A **21**, 1977-81

The contents of chapter 5, chapter 6 and the solution of $SO_3$ in chapter 7 have been published in

Searle B.G. and Butler P.H., 1988, J Phys A **21**, 3041-50

The solution of $K_{20}$ and the discussion of mixed symmetry triads from chapter 3 has been published in

Searle B.G. and Butler P.H., 1988, J Phys A **21**, 3313-19

These three papers are attached to this appendix.

J. Phys. A: Math. Gen. **21** (1988) 1977–1981. Printed in the UK

# Recursive calculation of transformation factors in terms of primitive factors

Barry G Searle and Philip H Butler
Physics Department, University of Canterbury, Christchurch, New Zealand

Received 22 December 1987

**Abstract.** We present an algorithm for recursively calculating coupling, recoupling, induction and similar coefficients in terms of the primitive coefficients. The algorithm is shown to be complete for any compact group. We use the 6-$j$ symbol as an example for describing our algorithm, but the algorithm applies to a wide class of group theoretic transformation factors.

## 1. Introduction

Group theory is of significant interest to physicists as a way of examining the symmetry of a system. The coupling of states for a particular group–subgroup pair to produce a new state introduces us to coupling coefficients and their associated recoupling coefficients. In a similar fashion, the induction of a subgroup into a group leads to the induction and reinduction factors (see Haase and Butler 1984). Finally the transformation from one lineal descent of subgroups of a group to a different set of subgroups requires knowledge of transformation coefficients. Hence it is useful to have a completely general method of calculating the values of the various symmetrising factors for any group. At present no complete algorithm exists that is applicable to all the point groups and to all the classical Lie groups, even for 6-$j$ and 3-$jm$.

Much of the recent work on particular groups has been for relatively simple multiplicity-free cases (see Judd 1986, 1987, Judd *et al* 1986, Haase and Dirl 1986) involving couplings of a small faithful irrep, which we shall call the primitive irrep. We will show that it is possible to calculate recursively any of the above factors for any other coupling, induction, recoupling or like process, in terms of the corresponding primitive coefficients. We prove the completeness of the algorithm, and show that it is unaffected by any multiplicity that occurs. The present algorithm reduces the problem of calculating a set of transformation factors to the problem of calculating the primitive factors. The primitive 6-$j$ occur in the recursion relations for non-primitive 6-$j$ and 3-$jm$ and will be considered in a forthcoming paper.

In § 2 we will define the conventions and equations that are required (see, e.g., Butler 1981, Bickerstaff *et al* 1982) and give a short review of an earlier more restricted algorithm for 6-$j$ and 3-$jm$ symbols. In § 3 we will describe our algorithm for a particular case, the 6-$j$ symbol, and in § 4 we comment on the application of our algorithm to other kinds of transformation factors.

0305-4470/88/091977+05$02.50  

## 2. Definition and review

A review of the coupling coefficients for an arbitrary compact group can be found in Butler (1975) and an introduction to induction factors in Haase and Butler (1984). In their earlier algorithm for calculating 6-*j* and 3-*jm* Butler and Wybourne (1976) defined a few terms that we will also use. The primitive irrep $\varepsilon$ is any low-dimension faithful irrep of the group that we choose for the role of being the pivotal irrep for the recursive building up of transformation factors. With respect to such a choice, the power $p(\lambda)$ of a general irrep $\lambda$, is defined as the smallest value of $k$ such that $(\varepsilon+\varepsilon^*)^k \supset \lambda$. One has $p(\lambda^*)=p(\lambda)$. An irrep $\lambda$ is considered to be less than another $\mu$ if $p(\lambda)<p(\mu)$.

We recall that a triad consists of three irreps $\lambda_1$, $\lambda_2$, $\lambda_3$ and a multiplicity index $r$, where the triad exists if the triple product $\lambda_1\times\lambda_2\times\lambda_3$ contains at least $r$ copies of the scalar irrep. We will use $n_{\lambda_1\lambda_2\lambda_3}$ for the maximum value of $r$ for which the triad exists. A trivial triad is one where one of the irreps is the scalar. Similarly a primitive triad is one that contains the primitive irrep $\varepsilon$ (or $\varepsilon^*$). No further distinction is made between the triads here, although it can be done. We will often use the symbol $\bar{\lambda}$ to denote any irrep such that both $p(\bar{\lambda})=p(\lambda)-1$ and the triad $\lambda\bar{\lambda}\varepsilon r$ exists for $r=1$.

A trivial or primitive 6-*j* or 3-*jm* symbol is defined as one which contains a trivial or primitive triad, and hence the scalar or primitive irrep. (For 3-*jm* symbols, we are only concerned with the nature of the group triad, not the subgroup triad.) In the rest of this paper we will only consider the calculation of those symbols that are neither primitive nor trivial, i.e. ones which are composed only of general irreps.

Butler and Wybourne (1976) proposed a method for recursively calculating 3-*jm* and 6-*j* symbols of this non-primitive type. They combined the orthogonality equation for the symbol with either the Wigner equation (for 3-*jm*) or the Biedenharn–Elliott equation (for 6-*j*) in such a way that a general symbol was written as a product of other general symbols whose smallest irrep was one power smaller than the smallest in the original. For example, the Wigner equation was modified to

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{pmatrix}^{r_4}_{s_4}$$

$$= \sum_{b_1\rho_1 b_2\rho_2 b_3\rho_3 s_1 s_2 s_3 r_2 r_3 \mu_1} |\lambda_1||\mu_1|(\mu_1)_{b_1\rho_1 b_1^*\rho_1^*}(\mu_2)_{b_2\rho_2 b_2^*\rho_2^*}(\mu_3)_{b_3\rho_3 b_3^*\rho_3^*}$$

$$\times \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \bar{\lambda}_1 & \varepsilon \end{Bmatrix}_{r_1 r_2 r_3 r_4} \begin{pmatrix} \lambda_1 & \bar{\lambda}_1 & \varepsilon \\ a_1\sigma_1 & b_2^*\rho_2^* & b_3\rho_3 \end{pmatrix}^{r_1}_{s_1} \begin{pmatrix} \mu_1 & \lambda_2 & \varepsilon^* \\ b_1\rho_1 & a_2\sigma_2 & b_3^*\rho_3^* \end{pmatrix}^{r_2}_{s_2}$$

$$\times \begin{pmatrix} \mu_1^* & \bar{\lambda}_1^* & \lambda_3 \\ b_1^*\rho_1^* & b_2\rho_2 & a_3\sigma_3 \end{pmatrix}^{r_3}_{s_3} \begin{Bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \\ \rho_1 & \rho_2 & \rho_3 \end{Bmatrix}_{s_1 s_2 s_3 s_4}.$$

In this way it was possible to recurse, with successive steps reducing the power of the smallest irrep until it was of power one. At this point the unknown symbol became a product of primitive ones, and no further recursion by this method was possible. The disadvantage of this is the extra sum over irrep $\mu$ when compared to the unmodified equation. This meant that although one irrep is being reduced, another irrep of the group is growing at the same rate (and hence the size of irreps that are branched to is also on the increase for the 3-*jm*) so that a fairly small general symbol required knowledge of the value of some fairly large primitive symbols to be able to get an answer.

### 3. Improved algorithm for 6-*j* symbols

For the purpose of calculating a 6-*j*, we arrange our unknown general 6-*j* by application of the various symmetries so that the smallest irrep is in the top right-hand corner of the 6-*j*. Denoting the 6-*j* in this arrangement as

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r} \tag{3.1}$$

we may use the Biedenharn–Elliott sum rule to give sufficient linearly independent linear equations for a set of such 6-*j*, so that we may solve for all $n_{\lambda_1\lambda_2\lambda_3}$ values of $r$. We choose the coefficients for these $n_{\lambda_1\lambda_2\lambda_3}$ unknown 6-*j* to be the primitive 6-*j* of the form

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3^* & \varepsilon & \nu \end{Bmatrix}_{s_1s_2s_3r}. \tag{3.2}$$

The Biedenharn–Elliott sum rule in the absence of mixed symmetry couplings (which in no way affect this algorithm) and with these coefficients is then

$$\begin{aligned} \sum_r \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3^* & \varepsilon & \nu \end{Bmatrix}_{s_1s_2s_3r} &= \sum_{\lambda t_1t_2t_3} |\lambda|\{\lambda_1\}\{\mu_1\}\{\lambda_1\varepsilon^*\nu s_1\}\{\bar{\lambda}_3^*\lambda_2\nu^* s_2\}\{\varepsilon\bar{\lambda}_3\lambda_3 s_3\} \\ &\quad \times\{\lambda^*\mu_1\bar{\lambda}_3 t_1\}\{\lambda\mu_2^*\varepsilon t_2\}\{\lambda\mu_3^*\nu t_3\} \\ &\quad \times \begin{Bmatrix} \nu^* & \lambda_2 & \bar{\lambda}_3^* \\ \mu_1 & \lambda & \mu_3 \end{Bmatrix}_{t_3r_2t_1s_2} \begin{Bmatrix} \lambda_1 & \nu & \varepsilon^* \\ \lambda & \mu_2 & \mu_3 \end{Bmatrix}_{r_1t_3t_2s_1} \begin{Bmatrix} \varepsilon & \bar{\lambda}_3 & \lambda_3 \\ \mu_1 & \mu_2 & \lambda \end{Bmatrix}_{t_2t_1r_3s_3}. \end{aligned} \tag{3.3}$$

For the point groups and for the classical Lie groups no multiplicity occurs for any primitive triad given the usual choice of $\varepsilon$, except in non-stretched couplings of the type $\varepsilon\times\lambda\supset\mu$, where $p(\lambda)=p(\mu)$. Non-stretched primitive couplings occur only for a few groups. The first 6-*j* on the right-hand side involves $\bar{\lambda}_3$ and thus may be regarded as preceding the unknown 6-*j* (which has $\lambda_3$ as its smallest entry), and the other two 6-*j* are primitive. This equation may be used recursively to reduce the power of the smallest irrep by one in the non-primitive 6-*j* until $\bar{\lambda}_3$ is of power one, and then all 6-*j* on the right-hand side are primitive.

If the multiplicity $n_{\lambda_1\lambda_2\lambda_3}$ is greater than one, then we have more than one unknown 6-*j*. It is always possible to select sufficient values of $s_1$, $s_2$ and $\nu$, to create $n_{\lambda_1\lambda_2\lambda_3}$ linearly independent equations which can be solved for the set of unknown 6-*j*. The proof of this is given in the following paragraph.

The set of 6-*j* in (3.2) formed by varying $\lambda_3s_3r$ as a row index and $\nu s_1s_2$ as a column index forms a square matrix with elements $S^{\lambda_3s_3r}_{\nu s_1s_2}$, since it is related by a column interchange to the unitary matrix $R$ of recoupling coefficients (Butler 1975, equation (9.13)).

$$R^{\lambda_3s_3r}_{\nu s_1s_2}=\langle(\lambda_1\varepsilon^*)s_1\nu^*,\bar{\lambda}_3,s_2\lambda_2^*|\lambda_1(\varepsilon\bar{\lambda}_3)s_3\lambda_3r\lambda_2^*\rangle. \tag{3.4}$$

This unitarity is expressed by the orthonormality equations for 6-*j*

$$\sum_{\lambda_3s_3r}|\lambda_3||\mu_3|\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r}\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \nu \end{Bmatrix}^*_{s_1s_2s_3r}=\delta_{\mu_3\nu}\delta_{r_1s_1}\delta_{r_2s_2} \tag{3.5}$$

obtained, for example, from Butler (1975, equation (9.11)) by a row flip. The consequence of this is that $S^{\lambda_3 s_2 r}_{\nu s_1 \bar{s}_2}$ is a non-singular square matrix, which implies that its columns are linearly independent. Hence a subset of rows for fixed $\lambda_3$ and $s_3$, a subset with $n_{\lambda_1\lambda_2\lambda_3}$ numbers, can be found that are linearly independent. The part of the matrix $S$ that we are using has $n_{\lambda_1\lambda_2\lambda_3}$ linearly independent rows, and so varying the indices $\nu_1 s_1$ and $s_2$ over the appropriate range will always give sufficient linear equations to allow the unknown 6-*j* of (3.1) to be solved.

Often the range of $\nu$ can be restricted, so as not to use all terms in the product $\varepsilon \times \lambda_1$. For example, one can usually, but not always, restrict $\nu$ to being $\bar{\lambda}_1^*$. This restriction is where our method has an advantage over that due to Butler and Wybourne (1976) who used the orthogonality of the 6-*j* to rewrite (3.3) in cases with multiplicity. Our algorithm only uses a few of the possible values of $\nu$ to solve all $n_{\lambda_1\lambda_2\lambda_3}$ unknown 6-*j*, whilst their algorithm effectively required a sum over all values of $\nu$ for each of the unknown 6-*j*. The maximum power of $\lambda$ that occurs in (3.3) is $p(\mu_2)+1$ since it occurs in the triad $(\lambda\mu_2^*\varepsilon)$, whereas with the previous algorithm two summations occur with maximum powers of $p(\mu_1)+1$ and $p(\mu_2)+1$ respectively. We therefore note that the most efficient use of equation (3.3) occurs when the unknown 6-*j* (3.1) is arranged so that the smallest of the irreps $\lambda_1\lambda_2\mu_1\mu_2$ is $\mu_2$. The largest possible term on the right-hand side of (3.3) is then the 6-*j* with powers

$$\begin{bmatrix} p(\lambda_1)+1 & p(\lambda_2) & p(\lambda_3)-1 \\ p(\mu_1) & p(\mu_2)+1 & p(\mu_3) \end{bmatrix}.$$

The study of this algorithm was initiated by a request from Hamer (see Hamer *et al* 1986) for 6-*j* of $SU_3$ beyond those of previously published tables (see Haase and Butler 1985, Bickerstaff *et al* 1982 and references therein). In calculating these 6-*j* of up to power 5 it became apparent that the algorithms we were using were not very efficient, and could be improved.

## 4. Applying the algorithm to other transformation factors

The algorithm presented can be applied to a variety of symbols that obey an orthornormality condition and which obey an equation involving the product of a factor and another equation being equal to a sum over three others. The Wigner equation for 3-*jm* symbols of a group G restricted to a subgroup H,

$$\begin{aligned} &\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{pmatrix}^{r_4}_{s_4} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r_4} \\ &\quad = \sum_{b_1\rho_1 b_2\rho_2 b_3\rho_3 s_1 s_2 s_3} (\mu_1)_{b_1\rho_1 b_1^*\rho_1^*} (\mu_2)_{b_2\rho_2 b_2^*\rho_2^*} (\mu_3)_{b_3\rho_3 b_3^*\rho_3^*} \\ &\qquad \times \begin{pmatrix} \lambda_1 & \mu_2^* & \mu_3 \\ a_1\sigma_1 & b_2^*\rho_2^* & b_3\rho_3 \end{pmatrix}^{r_1}_{s_1} \begin{pmatrix} \mu_1 & \lambda_2 & \mu_3^* \\ b_1\rho_1 & a_2\sigma_2 & b_3^*\rho_3^* \end{pmatrix}^{r_2}_{s_2} \\ &\qquad \times \begin{pmatrix} \mu_1^* & \mu_2 & \lambda_3 \\ b_1^*\rho_1^* & b_2\rho_2 & a_3\sigma_3 \end{pmatrix}^{r_3}_{s_3} \begin{Bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \\ \rho_1 & \rho_2 & \rho_3 \end{Bmatrix}_{s_1s_2s_3s_4} \end{aligned} \tag{4.1}$$

is a typical example of this type of equation. In fact the Biedenharn–Elliott equation is rather atypical since the coefficient of the 6-*j* on the left-hand side is itself a 6-*j*.

If the unknown 3-*jm* in this equation is arranged so that $\lambda_3$ is the smallest of the irreps in the group triad $\lambda_1\lambda_2\lambda_3 r_4$, then the 6-*j* for the group G that we use as coefficients are identical to those in (3.2). We note that the 6-*j* on the right-hand side of equation (4.1) is a 6-*j* for the subgroup H and for the present calculation may be regarded as a set of known quantities. The algorithm then proceeds recursively as before, with the eventual result being a dependence of the unknown on primitive 3-*jm*.

The method can also be applied directly to the induction factors since they obey a relation very similar to the relation between coupling and recoupling coefficients that leads to the Wigner equation (see Haase and Butler 1984, equation (5.7)). The reinduction factors that occur with the induction factors are very similar to recoupling factors and obey a relation very similar to the Biedenharn–Elliott equation. As a result it is trivial to apply the present algorithm to get a general induction (or reinduction) factor in terms of primitive induction (reinduction) factors.

## 5. Conclusion

The algorithm of Butler and Wybourne (1976) for calculating all vector coupling and recoupling coefficients for any compact group in terms of primitive coupling coefficients has been substantially improved for cases with coupling multiplicity. The algorithm has also been shown to apply quite generally to a wide category of transformation factors including the induction factors of Haase and Butler (1984). Being able to solve all general factors in terms of the primitive factors leaves us with the problem of solving the primitive factors. Such a calculation requires that certain phase and multiplicity choices be made.

## References

Bickerstaff R P, Butler P H, Butts M B, Haase R W and Reid M F 1982 *J. Phys. A: Math. Gen.* **15** 1087–117
Butler P H 1975 *Phil. Trans. R. Soc.* A **227** 545–85
—— 1981 *Point Group Symmetry Applications: Method and Tables* (New York: Plenum)
Butler P H and Wybourne B G 1976 *Int. J. Quant. Chem.* **10** 581–98
Haase R W and Butler P H 1984 *J. Phys. A: Math. Gen.* **17** 47–59
—— 1985 *J. Math. Phys.* **26** 1493–513
Haase R W and Dirl R 1986 *J. Math. Phys.* **27** 900–13
Hamer C J, Irving A C and Preece T E 1986 *Nucl. Phys.* B **270** 553–74
Judd B R 1986 *J. Math. Phys.* **27** 2616–22
—— 1987 *J. Phys. A: Math. Gen.* **20** L343–7
Judd B R, Lister G M S and O'Brien M C M 1986 *J. Phys. A: Math. Gen.* **19** 2473–86

J. Phys. A: Math. Gen. **21** (1988) 3041–3050. Printed in the UK

# Calculation of primitive 6-*j* symbols

Barry G Searle and Philip H Butler
Physics Department, University of Canterbury, Christchurch, New Zealand

Received 1 March 1988, in final form 19 April 1988

**Abstract.** The 6-*j* symbols of a group are independent of the subgroup chain chosen to define the basis states. We present an improved algorithm for calculating the primitive 6-*j* symbols for a compact group with a faithful irrep by recursive building up using only the Kronecker product rules and the general properties of 6-*j*. Previously one has sometimes needed to search for the useful equations by systematically trying all equations which involve unknown 6-*j*. We show that the primitive 6-*j* may all be easily solved in terms of a subclass, the core 6-*j*. We discuss how the core 6-*j* can usually be solved, proving that the method is complete for $SO_3$. We conjecture that the algorithm is complete for all groups.

## 1. Introduction

There continues to be interest in finding improved methods for calculating 3-*jm* and 6-*j* symbols for various groups, because of the use of many different groups in quantum mechanics. Some recent examples of such calculations are Chen *et al* (1985) for the space groups, Haase and Dirl (1986) for the symmetric groups, Judd (1986, 1987), Judd *et al* (1986) and Pluhar *et al* (1986) for the classical groups, Raynal and Conte (1985) for the point groups and Zeng (1987) for OSp(1, 2).

Most authors have calculated 3-*jm* from explicitly symmetrised basis functions and then calculated 6-*j* from the 3-*jm*. Using such an approach, a basis choice for the partners of each irrep must be made to calculate the 3-*jm*, even though the 6-*j* are totally independent of such a choice. However when calculating 6-*j*, an alternative and more direct approach is to use the Biedenharn–Elliott, the Racah backcoupling and the orthonormality relations of 6-*j*, relations which are valid for all compact groups (see Derome and Sharp 1965, Butler 1975). As well as these generally valid relations, we require specific information about the group, specifically the product rules and plethysms (or symmetrised powers) of the irreps. An early example of such a calculation is the demonstration (Butler 1976) that the Racah formula for the 6-*j* of $SO_3$ can be obtained directly from the product rules of $SO_3$. If one requires 3-*jm*, a basis choice can then be made using the appropriate subgroups, and the general 3-*jm* can be calculated from this information and the 6-*j* of the group and subgroup. Complete tables for all point groups (up to $j = 8$) have been calculated by this means (Butler 1981) and we have further improved upon that algorithm for all non-primitive transformation factors in a previous paper (Searle and Butler 1988).

Judd *et al* (1986) emphasise that calculating multiplicity-free 6-*j* for a general compact group is a relatively simple extension of the methods for $SO_3$. Our preceding paper showed that the calculation of non-primitive 6-*j* symbols with multiplicity may

0305-4470/88/143041+10\$02.50  

be carried out essentially as for the multiplicity-free case once the primitive symbols have been chosen. This paper will consider the calculation of primitive 6-*j* by placing them into various classes, and recursing downward until we reach a core 6-*j*. The calculation of the core 6-*j* includes calculation of basis 6-*j* which requires the choice of the free phases in the 6-*j* algebra and the separation of the coupling multiplicities.

The present study grew out of a request by Hamer (see Hamer *et al* 1986) for some 6-*j* of $SU_3$ involving irreps up to power five. The ALGOL program used by Butler (1981) for calculating 6-*j* and 3-*jm* of the point groups, and used by Bickerstaff *et al* (1982) for some 6-*j* of $SU_3$ and $SU_6$, was found to be unnecessarily indirect. It required more intermediate steps than were necessary and included the calculation of 6-*j* outside the range of interest. This arises partly because $SU_3$ contains the case $\{1\}\times\{1\}\supset\{1\}^*$, a case which does not occur when the primitive irrep is symplectic and indeed occurs only rarely amongst the classical groups. Our attempts to avoid the need to calculate 6-*j* outside Hamer's range of interest led us to the present algorithm.

Section 2 reviews the definitions needed in this paper. Section 3 discusses the number and occurrence of phase choices available in the 6-*j* part of the Racah–Wigner algebra, results due to Derome (1966) and used by him to make certain advantageous phase choices in the symmetry relations. In § 4 we categorise a subset of the 6-*j* as the basis set of 6-*j*. This basis set may be used to fix all the free phases and multiplicity separations of the 6-*j* algebra. Section 5 splits the primitive 6-*j* into various subclasses and shows how to solve for 6-*j* in some of the classes in terms of 6-*j* of a class of core 6-*j*. All the basis 6-*j* belong to the core subclass. It is the discovery of this classification and the consequential use of the recursion relations that is the central result of this paper. Our building-up method in the past has always provided sufficient relations for the 6-*j* of all the various groups we have studied. However one has sometimes needed to search for the useful equations by the tedious method of systematically trying all equations which involve the unknown 6-*j*. Section 5 proves that the primitive 6-*j* may all be written in terms of the core subclass. As noted above, this class of core 6-*j* includes all basis 6-*j*, and in § 6 we discuss how 6-*j* in this class can usually be solved.

The problem of calculating the core 6-*j* remains partly open. The group $SO_3$ is special in having one irrep of each power, and this allows us to prove completeness for $SO_3$. Likewise explicit calculation shows that the 6-*j* for all the point groups and $SU_3$ are readily calculated by the present algorithm.

## 2. Definitions and reviews

Derome and Sharp (1965) introduced the matrix $m(\pi, \lambda_1\lambda_2\lambda_3)$, indexed by coupling multiplicity labels, to describe the symmetries under column permutations, $\pi$, of a 3-*jm* symbol involving the irreps $\lambda_1\lambda_2\lambda_3$ of a compact group. The same paper introduced a generalisation of the 6-*j* symbols of angular momentum that give the remaining information on recoupling transformations in the Racah–Wigner algebra. Butler (1975) gives a review of the Racah–Wigner algebra for the case of a general compact group, while Butler (1981) gives an account of the definitions and results appropriate to those couplings, such as those within the point groups, that have a simple permutation symmetry. (The more general case occurs when the coupling of three identical irreps to a scalar is said to be of mixed symmetry, the three irreps of the threefold coupling transforming amongst themselves as the irrep [21] of the symmetric group $S_3$.) The above references show that the familiar methods of angular momentum theory apply to any compact group subject to the generalisations required when: (i) an irrep $\lambda$ is not unitarily equivalent to its complex conjugate $\lambda^*$; (ii) when a coupling of the three irreps $\lambda_1\lambda_2\lambda_3$ gives the scalar irrep more than once; (iii) when mixed symmetry couplings occur.

As in our previous work (Butler 1981, Searle and Butler 1988) we use the concept of the power $p(\lambda)=k$ of an irrep $\lambda$, where $(\varepsilon+\varepsilon^*)^k \supset \lambda$ and $\varepsilon$ is the primitive irrep, to define a partial ordering: we say that $\lambda <_{\mathrm{p}} \mu$ if $p(\lambda)<p(\mu)$. The ordering of irreps is then arbitrarily completed ensuring only that $\lambda$ and $\lambda^*$ are contiguous irreps.

A triad $\lambda_1\lambda_2\lambda_3 r$ is defined to exist if $\lambda_1\times\lambda_2\times\lambda_3$ contains the scalar at least $r$ times. The triad is said to be in standard order if $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$. For the purposes of the classification we must define a partial ordering between two triads $\lambda_a\lambda_b\lambda_c r$, $\mu_a\mu_b\mu_c s$ by writing them in standard order $\lambda_1\lambda_2\lambda_3 r$ and $\mu_1\mu_2\mu_3 s$, and saying that $\lambda_1\lambda_2\lambda_3 r <_{\mathrm{p}} \mu_1\mu_2\mu_3 s$ if $p(\lambda_2)+p(\lambda_3)<p(\mu_2)+p(\mu_3)$. This ordering differs from the one used by us for tabulation purposes. The present partial ordering ($<_{\mathrm{p}}$) is to be completed by basing the further ordering on the completed ordering of the irreps. Further, a triad is in standard form if it is in both standard order and $\lambda_1\lambda_2\lambda_3 r < \lambda_1^*\lambda_2^*\lambda_3^* r$ in the completed ordering.

In addition to using these orderings, we classify triads by the power of their smallest irrep. A trivial triad contains the scalar irrep and is of the form $\lambda^*\lambda 0$, while a primitive triad contains an irrep of power one, that is, $\varepsilon$ or $\varepsilon^*$ (but we do not include the $\varepsilon^*\varepsilon 0$ triad). In the following we will often use $\varepsilon_1$, $\varepsilon_2$, etc, to denote an irrep of power one (either $\varepsilon$ or $\varepsilon^*$). A triad is said to be stretched if $p(\lambda_1)=p(\lambda_2)+p(\lambda_3)$ when in standard order. For all the (double covered) point groups, all symmetric groups and all simple compact Lie groups, these triads are of multiplicity one if $\varepsilon$ is chosen as the lowest-dimensional faithful irrep. For most of these groups the sets of primitive and stretched primitive triads are identical. However, for four of the above groups, namely $SU_3$, $E_6$, $E_8$ and $G_2$, the stretched primitive set is smaller because the non-stretched coupling $\varepsilon\times\varepsilon\supset\varepsilon^*$ occurs (if spinor irreps are excluded and the vector irrep is chosen as the primitive irrep such products also occur for some point groups, for all $SO_n$ and for $S_n$). We will often write a stretched primitive triad in the standard form as $\lambda\bar{\lambda}\varepsilon_1$, since there is no multiplicity, and we use the notation $\bar{\lambda}$ to indicate any irrep of power $p(\lambda)-1$ contained in either of the products, $\lambda^*\times\varepsilon^*$ or $\lambda^*\times\varepsilon$.

In similar fashion to the definition of triads, a trivial 6-*j* is defined as a 6-*j* that contains the scalar irrep. A primitive 6-*j* does not contain 0 but does contain $\varepsilon$ or $\varepsilon^*$ and hence contains at least two primitive triads. A core 6-*j* is a certain kind of primitive 6-*j* and will be defined in § 5.

## 3. The phase factor matrices

Derome (1966) exploited the unitary freedom in the multiplicity space of a coupling coefficient to analyse the possible structures of the permutation matrices $m(\pi, \lambda_1\lambda_2\lambda_3)$. If $\lambda_1\lambda_2\lambda_3$ couple together to give $n$ copies of the scalar irrep, so that there are $n$ non-vanishing triads $\lambda_1\lambda_2\lambda_3 r$, then there is a $n\times n$ unitary matrix $K(\lambda_1\lambda_2\lambda_3)$ describing the transformation between one set of coupling coefficients with a given set of phases and multiplicity separations, and another such set. For each triple $\lambda_1\lambda_2\lambda_3$ in standard order, there are up to twelve distinct phase freedom matrices, $K$, one for each of the six orders and one for each complex conjugate triple. One of the principal results of Derome (1966) was to show how to exploit the phase freedom matrices to select simple values of the elements of the matrix $m(\pi, \lambda_1\lambda_2\lambda_3)$. We call the matrix elements the 3-$j$ and write them as $\{\lambda_1\lambda_2\lambda_3\}_{rs}$.

Except for mixed symmetry couplings, the 3-$j$ may be chosen (or fixed, depending on the case) as diagonal in rs, $\pm1$ for interchanges ($+1$ for cyclic permutations) and independent of the order of the triad (Butler 1975). This choice leaves us the freedom of choosing one phase freedom matrix $K(\lambda_1\lambda_2\lambda_3)$ for each ordered triple $\lambda_1\lambda_2\lambda_3$. If any two of the irreps in the triple are identical, the phase freedom matrix is not totally free but is restricted to being block diagonal with respect to the permutation symmetry type. Butler (1975) also chooses the Derome and Sharp (1965) $A$ matrix to be unity, hence fixing the relationship of $K(\lambda_1^*\lambda_2^*\lambda_3^*)$ to $K(\lambda_1\lambda_2\lambda_3)^*$. (Bickerstaff and Damhus (1985) argue that a unit choice for $A$ may not be the most propitious choice; however the actual choice does not affect the results of this paper.) Any choice of a standard form for the $m$ and $A$ matrices fix the relationship of the up to twelve distinct phase freedom matrices $K$. In the following sections we may therefore consider the matrix $K(\lambda_1\lambda_2\lambda_3)$ free if and only if the triple $\lambda_1\lambda_2\lambda_3$ is in standard form.

The above discussion looked at the coupling phase freedom. A similar argument applies to the branching phase freedoms which occur in the 3-$jm$ part of the Racah–Wigner algebra. Bickerstaff (1984) and Bickerstaff and Damhus (1985) discuss conditions on the choice (or lack) of reality of coupling coefficients. However the coupling and branching phase choices are not always the only choices one must make within the Racah–Wigner algebra. Reid and Butler (1980, 1982) show that additional phases can occur for some group–subgroup pairs, phases that are related in some way to the orientation of the subgroup H within the group G. These orientation phases occur since certain basis kets are not fixed by the 3-$jm$ algebra, and involve special cases of the branching rules. It is possible that the product rules $\lambda_1\times\lambda_2\supset\lambda_3$ contain similar choices since they are equivalent to a branching $G\times G\supset G$. However, we have found no evidence that such phases occur in the 6-$j$ algebra of any of the groups we have studied.

## 4. The basis 6-$j$

Two alternative choices of the phase and multiplicity separation in a 6-$j$ symbol are related by four coupling phase freedom matrices in the following manner:

$$\begin{Bmatrix}\lambda_1 & \lambda_2 & \lambda_3\\ \mu_1 & \mu_2 & \mu_3\end{Bmatrix}^{\mathrm{alt}}_{s_1s_2s_3s_4} = K(\lambda_1\mu_2^*\mu_3)^{s_1}{}_{r_1}K(\mu_1\lambda_2\mu_3^*)^{s_2}{}_{r_2}K(\mu_1^*\mu_2\lambda_3)^{s_3}{}_{r_3}K(\lambda_1^*\lambda_2^*\lambda_3^*)^{s_4}{}_{r_4} \times\begin{Bmatrix}\lambda_1 & \lambda_2 & \lambda_3\\ \mu_1 & \mu_2 & \mu_3\end{Bmatrix}_{r_1r_2r_3r_4}. \tag{4.1}$$

The non-primitive 6-$j$ have previously been shown (Searle and Butler 1988) to be totally dependent on the primitive 6-$j$ and hence contain no freedom at all. This means that the freedom allowed by the $K$ matrices is only constrained by choices of primitive 6-$j$.

Each irrep in a 6-$j$ occurs in two triads. As we build up our 6-$j$ we will at some time try to solve for a 6-$j$ which contains an irrep, $\lambda$, that has not previously been used. Since this irrep occurs in two triads, we have two new free phase matrices arising, one for each triad. We are allowed to choose the phase of the 6-$j$ if the two triads are distinct. Such a phase choice results in the choice of (part of—if there is multiplicity) one of the two free $K$ matrices involving $\lambda$ with respect to (part of) the other and with respect to the $K$ matrices of the other two triads in the 6-$j$. This process occurs for every triad that has the irreps $\lambda$ or $\lambda^*$ as the largest irrep, resulting in all the phases being chosen with respect to the one matrix $K$ that is still free. We arbitrarily choose the triad that has the free phase matrix to be one of the stretched primitive triads $\lambda\bar{\lambda}\varepsilon_1$ for the irrep $\lambda$. This matrix $K(\lambda\bar{\lambda}\varepsilon_1)$ cannot be fixed within the 6-$j$ algebra (Bickerstaff 1981) and the triad is known as the basis triad for the pair $\lambda$, $\lambda^*$. Those 6-$j$ for which a phase is chosen are said to be the basis 6-$j$ corresponding to those triads whose phase they fix. This selection process occurs once for every irrep $\lambda$ (or pair $\lambda, \lambda^*$) where $\lambda > \varepsilon$. In groups where $\varepsilon \times \varepsilon \supset \varepsilon^*$ we find that the phase matrix of the triad $\varepsilon\varepsilon\varepsilon$ cannot be fixed by the algebra either, so that the primitive irrep has a basis triad in these few cases.

For each triad $\lambda\alpha\beta r_4$ where $p(\beta) > 1$ and where the triad is in standard form, we will select as the basis 6-$j$ the 'least' of the 6-$j$ in the form

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \beta' & \varepsilon_1 & \lambda' \end{Bmatrix}_{r_1 r_2 r_3 r_4}. \tag{4.2}$$

We define this 'least' 6-$j$ by choosing $\varepsilon_1 = \varepsilon$ and the irreps $\beta'$ and $\lambda'$ to be as small as possible in the following manner. Sometimes there is more than one irrep $\beta'$ of power $p(\beta) - 1$ that can be used. In a similar way several $\lambda'$ (and $r_2$) may occur. We select the smallest irrep $\beta'$ for which the power of $\lambda'$ is a minimum, and the smallest irrep $\lambda'$ of this minimum power and where the 6-$j$ is non-zero (see § 6). Finally $\varepsilon^*$ may be used instead of $\varepsilon$ if it allows irreps of smaller power to be chosen. Usually the resulting basis 6-$j$ is of the form

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \bar{\beta} & \varepsilon & \bar{\lambda} \end{Bmatrix}_{r_1 r_2 r_3 r_4}$$

and has two non-primitive triads $\lambda\alpha\beta r_4$, $\bar{\lambda}^*\alpha\bar{\beta} r_2$.

When more than one irrep $\bar{\lambda}$ exists for a given $\lambda$, then there is more than one primitive triad for the irrep (of the form $\lambda\bar{\lambda}\varepsilon$ or $\lambda\bar{\lambda}\varepsilon^*$). Any one of these triads can be chosen basis. We choose the one with the smallest $\bar{\lambda}$. The remaining primitive triads give rise to corresponding basis 6-$j$. If we choose $\beta' = \bar{\beta}$ as in the form above we will find that $\bar{\beta} = 0$. We cannot choose this as a basis 6-$j$ as this is a trivial 6-$j$ with no freedom of phase. For most groups the smallest useful value for $\beta'$ is one of the power two irreps. We choose the smallest value of $\lambda'$ that results in distinct triads involving $\lambda$, thus ensuring that there is sufficient freedom in the 6-$j$ for it to be basis. The resulting basis 6-$j$ is of the form

$$\begin{Bmatrix} \lambda & \bar{\lambda} & \varepsilon_1 \\ 2 & \varepsilon_2 & \bar{\lambda}_b \end{Bmatrix}_{r_1 r_2 r_3 r_4} \tag{4.3}$$

where $\lambda\bar{\lambda}_b\varepsilon_2^*$ is the triad chosen as basis for $\lambda$. For those few groups where $\varepsilon\varepsilon\varepsilon$ is a triad, triads of the form $\lambda\alpha\varepsilon_1$ occur where $p(\alpha) = p(\lambda)$ or $p(\alpha) = p(\lambda) - 1$. In either case we have always found a suitable $\beta' = \varepsilon_3$ and $\lambda' = \bar{\lambda}$ to give a basis 6-$j$ of the form

$$\begin{Bmatrix} \lambda & \alpha & \varepsilon_1 \\ \varepsilon_3 & \varepsilon_2 & \bar{\lambda} \end{Bmatrix}_{r_1 r_2 r_3 r_4}. \tag{4.4}$$

## 5. Primitive 6-$j$ symbols

In this section we classify all primitive 6-$j$ into various forms determined by the number and the position of the primitive irreps. We then use the Racah backcoupling and Biedenharn–Elliott equations as recursion relations to relate any primitive 6-$j$ to 6-$j$ in one particular form and primitive 6-$j$ involving only smaller triads.

Consider first the primitive 6-$j$ with at least one non-primitive triad. In order to classify these 6-$j$ we use symmetry relations to rearrange the 6-$j$ so that the largest non-primitive triad $\lambda\alpha\beta r_4$ is in standard order and in the top row. For those primitive 6-$j$ with two primitive triads (so two triads are non-primitive) we obtain a 6-$j$ in one of three general forms, depending on the position in the bottom row of the primitive irrep. The primitive 6-$j$ with one non-primitive triad can be rearranged to have two primitive irreps in the bottom row. Those 6-$j$ with a primitive irrep in column 2 will be defined as core 6-$j$,

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \beta' & \varepsilon_1 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \alpha & \beta \\ \beta' & \varepsilon_1 & \varepsilon_2 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \alpha & \beta \\ \varepsilon_2 & \varepsilon_1 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.1}$$

while the others are not:

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \alpha' & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \alpha & \beta \\ \varepsilon_2 & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.2}$$

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \varepsilon_1 & \beta' & \alpha' \end{Bmatrix}_{r_1r_2r_3r_4}. \tag{5.3}$$

In (5.1), (5.2) and (5.3) the coupling conditions restrict the powers of $\alpha'$, $\beta'$ and $\lambda'$ to be within one of the powers of $\alpha$, $\beta$ and $\lambda$ respectively.

All other primitive 6-$j$ contain four primitive triads and may be related by the symmetry relations to one of the following:

$$\begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \varepsilon_3 & \varepsilon_1 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \varepsilon_3 & \varepsilon_1 & \varepsilon_4 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \varepsilon_5 & \varepsilon_2 \\ \varepsilon_3 & \varepsilon_1 & \varepsilon_4 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \\ \varepsilon_4 & \varepsilon_5 & \varepsilon_6 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.4}$$

$$\begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \alpha' & \varepsilon_3 & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \varepsilon_3 & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \begin{Bmatrix} \lambda & \varepsilon_4 & \varepsilon_2 \\ \alpha & \varepsilon_3 & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.5}$$

$$\begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \alpha' & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} \tag{5.6}$$

where $\lambda\alpha\varepsilon_2$ is the largest triad (so $\lambda \geqslant \alpha$, $\alpha \geqslant \lambda'$ and $\lambda \geqslant \alpha'$). The 6-$j$ of (5.4) only occur for the groups where $\varepsilon\varepsilon\varepsilon$ exists. It is easy to show that none of the irreps $\lambda$, $\alpha$ and $\lambda'$ can be of power greater than three in (5.5). We will define the 6-$j$ in (5.4) and (5.5) to be part of the set of core 6-$j$.

For most groups (that is where the triad $\varepsilon\varepsilon\varepsilon$ does not exist) all basis 6-$j$ belong to one of the forms of (5.1). However, to be consistent with the above classification the 6-$j$ in (4.3) must be related by symmetry to a 6-$j$ in the form

$$\begin{Bmatrix} \bar{\lambda} & \bar{\lambda}_b & 2 \\ \varepsilon_2 & \varepsilon_1 & \lambda' \end{Bmatrix}_{r_1r_2r_3r_4}$$

since $\bar{\lambda}\bar{\lambda}_b 2r_4$ is the only non-primitive triad. For the few groups that contain $\varepsilon\varepsilon\varepsilon$, the basis 6-$j$ in (4.2) belong to one of the forms of (5.1), whereas those in (4.4) belong to the form of (5.4).

We refer to the 6-*j* in (5.1), (5.4) and (5.5) as the core 6-*j*, and emphasise that our set of basis 6-*j* is a subset of the set of core 6-*j*. In the remainder of this section we shall give a recursive method of solving all non-core 6-*j*.

Any 6-*j* in the form of (5.3) may be related to 6-*j* in the other primitive forms by applying the Racah backcoupling relation (Butler 1975, equation (9.10)), together with the symmetry relations. The first 6-*j* that occurs is in the form of one in (5.1), and the second 6-*j* is of the form of one in (5.2). After application of the symmetries, we have

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \varepsilon_1 & \beta' & \alpha' \end{Bmatrix}_{r_1r_2r_3r} = \sum_{\substack{\rho=\lambda-1 \\ s_1s_2}}^{\lambda+1} \# \begin{Bmatrix} \lambda & \alpha & \beta \\ \beta'^* & \varepsilon_1^* & \rho \end{Bmatrix}_{s_1s_2r_3r} \begin{Bmatrix} \rho^* & \alpha & \beta' \\ \alpha'^* & \lambda & \varepsilon_1^* \end{Bmatrix}_{s_1r_2r_1s_2} \tag{5.7}$$

where $\lambda \pm 1$ denotes all irreps of power $p(\lambda) \pm 1$. Since we are only interested in the form of the equations for the 6-*j*, we have put all phase and dimension factors, in this equation and those that follow, into the symbol $\#$.

The Biedenharn–Elliott equation, when applied to a 6-*j* from (5.2), gives

$$\sum_r \begin{Bmatrix} \lambda & \alpha & \beta \\ \alpha' & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r} \begin{Bmatrix} \lambda & \alpha & \beta \\ \bar{\beta} & \varepsilon_2 & \nu \end{Bmatrix}^*_{s_1s_2s_3r}$$
$$= \sum_{\substack{\rho=\lambda'-1 \\ t_1t_2t_3}}^{\lambda'+1} \# \begin{Bmatrix} \lambda'^* & \rho & \varepsilon_2 \\ \nu & \lambda^* & \varepsilon_1 \end{Bmatrix}_{r_2t_1s_1t_2} \begin{Bmatrix} \nu^* & \alpha & \bar{\beta} \\ \alpha' & \rho & \varepsilon_1 \end{Bmatrix}_{t_1r_2t_3s_2} \begin{Bmatrix} \lambda'^* & \alpha' & \beta^* \\ \bar{\beta}^* & \varepsilon_2^* & \rho \end{Bmatrix}_{t_2t_3s_3r_3} \tag{5.8}$$

where the symmetries have again been used to rearrange the forms. The coefficient 6-*j* (second from the left) is in the form of the basis 6-*j* (and is basis when $\nu$ is a minimum), and we have shown in our paper on non-primitive coupling coefficients (Searle and Butler 1988) that the Biedenharn–Elliott equation will allow us to completely solve the set of unknowns when all $\nu$ are considered. The first 6-*j* on the right-hand side has only primitive triads and is in ~~one of (5.4)~~, (5.5) or (5.6). The second 6-*j* is the same form as the unknown but $\beta$ has been reduced to $\bar{\beta}$. Thus the largest non-primitive triad has been reduced (often further gains are made because $\nu = \bar{\lambda}$ as well). The third 6-*j* is of the form (5.1), and is to be solved by other means.

The 6-*j* of (5.6), where $\lambda'$, $\lambda$, $\alpha'$ and $\alpha$ are all non-primitive, may be related to 6-*j* that belong to those in (5.1) or (5.4) by use of the Racah backcoupling equation, where the maximum power of $\rho$ is two:

$$\begin{Bmatrix} \lambda & \alpha & \varepsilon_2 \\ \alpha' & \lambda' & \varepsilon_1 \end{Bmatrix}_{r_1r_2r_3r_4} = \sum_{\substack{\rho=0 \\ s_1s_2}}^{2} \# \begin{Bmatrix} \lambda'^* & \alpha^* & \rho^* \\ \varepsilon_2^* & \varepsilon_1^* & \lambda \end{Bmatrix}_{r_1r_4s_1s_2} \begin{Bmatrix} \lambda' & \alpha & \rho \\ \varepsilon_1^* & \varepsilon_2^* & \alpha'^* \end{Bmatrix}_{r_3r_2s_1s_2}. \tag{5.9}$$

This means that we have solved all primitive 6-*j* in terms of the core 6-*j*.

## 6. The core 6-*j*

A difficulty with finding a proof of completeness for our algorithm for the core 6-*j* is due to the fact that no single relation (orthonormality, Racah backcoupling or Biedenharn–Elliott sum rule) will always give sufficient linearly independent equations to solve for all unknowns. The set of core 6-*j* has the basis 6-*j* as a subset. The phase (and sometimes the multiplicity separation) of a basis 6-*j* is free so it can only be possible to find the magnitude of the unknown via equations which are quadratic. As a consequence we know that any complete set of suitable equations cannot be linear, so theorems for the completeness of linear equations are of little use.

When a free triad $\lambda\alpha\beta r$ exists for $r > 1$ we get one basis 6-*j* for each value of $r$ and need to resolve the multiplicity of these couplings subject to the group selection rules. When the triads have different symmetry types for some of the values of $r$, the equations that depend on the symmetry give extra information. When different values of $r$ have the same symmetry, symmetry will only partially solve the problem of separating the multiplicities and additional choices are required.

A second difficulty occurs if a 6-*j* chosen to be a basis 6-*j* turns out to be zero, because no phase is fixed for such a zero value. A 6-*j* is chosen to be the basis 6-*j* for a triad $\lambda\alpha\beta r$ depending on the value of $\beta'$ and $\lambda'$ (see § 4). If such a choice leads to a zero then the choice of $\lambda'$ or $\beta'$ must be revised, and we must solve for the magnitude of the revised choice of basis 6-*j* so as to fix the associated $K$ matrix. The occurrence of zero values is normal for a triad $\lambda\alpha\beta r$ with multiplicity when there is insufficient symmetry information to completely separate the multiplicity. In this case we must choose the separation, and a zero for the value of $r$ is the easiest choice to make, although this choice results in the phase of the corresponding triad remaining free until a revised value of $\beta'$ and $\lambda'$ is used for the given value of $r$.

The consequence of these problems is that it is difficult to enunciate a complete algorithm for the core 6-*j*, preventing us from proving that our algorithm is complete. However we have always been able to find sufficient equations to solve for any group we have so far attempted and we can prove completeness for $SO_3$.

$SO_3$ is the unique compact Lie group with one irrep of each power and it is multiplicity free. We use the example of $SO_3$ first to illustrate the kind of procedure to follow for typical groups and second to prove completeness in this special case. As with all groups that we have applied this algorithm to, we find that the orthonormality relation is sufficient to solve for almost all (but not all) of the core 6-*j*.

In $SO_3$ all primitive triads are stretched and there is only one primitive triad for each irrep so there are no basis 6-*j* for the case $\beta = \varepsilon$. Since there is only one irrep of each power the choice of $\beta'$ and $\lambda'$ in (4.2) is unique. We find that the 6-*j* in (5.4) do not occur and the set of 6-*j* in (5.5) reduce to

$$\begin{Bmatrix} \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \end{Bmatrix} \begin{Bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \end{Bmatrix}$$

whilst the 6-*j* with one non-primitive triad in (5.1) become, for all $\alpha \geqslant 1$,

$$\begin{Bmatrix} \alpha+1 & \alpha & 1 \\ \frac{1}{2} & \frac{1}{2} & \alpha+\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \alpha & \alpha & 1 \\ \frac{1}{2} & \frac{1}{2} & \alpha+\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \alpha & \alpha & 1 \\ \frac{1}{2} & \frac{1}{2} & \alpha-\frac{1}{2} \end{Bmatrix}.$$

We choose the first and last as basis 6-*j* for all triads $\alpha+1\alpha 1$ and $\alpha\alpha 1$ respectively.

The orthonormality equations give these basis 6-*j* immediately using the following equations (the summation is over the irreps of $SO_3$, not the powers):

$$\sum_{\rho=1 \text{ only}} \# \left| \begin{Bmatrix} \alpha+1 & \alpha & \rho \\ \frac{1}{2} & \frac{1}{2} & \alpha+\frac{1}{2} \end{Bmatrix} \right|^2 = 1$$

$$\sum_{\rho=0}^{1} \# \left| \begin{Bmatrix} \alpha & \alpha & \rho \\ \frac{1}{2} & \frac{1}{2} & \alpha-\frac{1}{2} \end{Bmatrix} \right|^2 = 1$$

where we recall that 6-*j* with the identity irrep are always known. We can solve for the above non-basis 6-*j* by using the following equations. The equations relate the unknowns to trivial 6-*j* and to the basis 6-*j* for the non-primitive triads that occur:

$$\sum_{\rho=0}^{1} \# \begin{Bmatrix} \frac{1}{2} & \frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & \rho \end{Bmatrix} \begin{Bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \rho \end{Bmatrix}^* = 0$$

$$\begin{Bmatrix} \frac{1}{2} & 1 & \frac{3}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \end{Bmatrix} = \sum_{\rho=0}^{1} \# \begin{Bmatrix} 1 & \frac{1}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & \rho \end{Bmatrix} \begin{Bmatrix} \frac{1}{2} & \frac{1}{2} & \rho \\ 1 & 1 & \frac{1}{2} \end{Bmatrix}$$

$$\sum_{\rho=0}^{1} \# \begin{Bmatrix} \alpha & \alpha & \rho \\ \frac{1}{2} & \frac{1}{2} & \alpha+\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \alpha & \alpha & \rho \\ \frac{1}{2} & \frac{1}{2} & \alpha-\frac{1}{2} \end{Bmatrix}^* = 0.$$

The remaining 6-*j* in (5.1) with two non-primitive triads have three possible forms:

$$\begin{Bmatrix} \lambda & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \lambda & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda+\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \lambda & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix}$$

where the first of these is chosen as the basis 6-*j* for $\lambda\alpha\beta$, and where $\beta > 1$ (as well as $\beta = 1$ for the third case). We may use the orthonormality equations to solve for these 6-*j*. The first two equations below relate the core 6-*j* to the basis 6-*j*, and the last is the normality relation required to find the magnitude of the basis 6-*j* (and is recursive in that it requires knowledge of smaller core 6-*j*),

$$\sum_{\beta'} \# \begin{Bmatrix} \lambda & \alpha & \beta' \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda+\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \lambda & \alpha & \beta' \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix}^* = 0$$

$$\sum_{\lambda'} \# \begin{Bmatrix} \lambda' & \alpha & \beta \\ \beta+\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix} \begin{Bmatrix} \lambda' & \alpha & \beta \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix}^* = 0$$

$$\sum_{\beta'} \# \left| \begin{Bmatrix} \lambda & \alpha & \beta' \\ \beta-\frac{1}{2} & \frac{1}{2} & \lambda-\frac{1}{2} \end{Bmatrix} \right|^2 = 1$$

where in each case the sum contains two terms.

In the above calculation for $SO_3$, we note that orthonormality is only once insufficient for a complete solution. For all groups that we have considered it is the core 6-*j* of low power that has caused us the most problems. $SO_3$ gives an example of this since it is a small core 6-*j* that cannot be solved by orthonormality.

## 7. Conclusion

We have significantly improved the algorithm for the calculation of primitive 6-*j*. By defining a subclass of primitive 6-*j* known as the core 6-*j* we have been able to give a complete method for calculating all non-core primitive 6-*j*. These core 6-*j* form a minority of the primitive 6-*j*, for example in the octahedral group (see the table on p 439 in Butler (1981)) there are 100 primitive 6-*j*, 45 of which are core (20 of these core 6-*j* are basis). A significant number of these core 6-*j* involve irreps of low power and seem to be the hardest to resolve.

We have been unable to give a complete algorithm to solve for the core 6-*j* for a compact group with faithful irreps, although we have been able to do so for $SO_3$ and have been able to resolve the problem for all groups we have so far considered.

## References

Bickerstaff R P 1981 *PhD thesis* University of Canterbury, New Zealand
—— 1984 *J. Math. Phys.* **25** 2808–14
Bickerstaff R P, Butler P H, Butts M B, Haase R W and Reid M F 1982 *J. Phys. A: Math. Gen.* **15** 1087–117
Bickerstaff R P and Damhus T 1985 *Int. J. Quantum Chem.* **27** 381–91
Butler P H 1975 *Phil. Trans. R. Soc.* A **277** 545–85
—— 1976 *Int. J. Quantum Chem.* **10** 599–613
—— 1981 *Point Group Symmetry Applications: Methods and Tables* (New York: Plenum)
Chen J-Q, Gao M-J and Ma G-Q 1985 *Rev. Mod. Phys.* **57** 211–78
Derome J R 1966 *J. Math. Phys.* **7** 612–5
Derome J R and Sharp W T 1965 *J. Math. Phys.* **6** 1584–90
Haase R W and Dirl R 1986 *J. Math. Phys.* **27** 900–13
Hamer C J, Irving A C and Preece T E 1986 *Nucl. Phys.* B **270** 553–74
Judd B R 1986 *J. Math. Phys.* **27** 2616–22
—— 1987 *J. Phys. A: Math. Gen.* **20** L343–7
Judd B R, Lister G M S and O'Brien M C M 1986 *J. Phys. A: Math. Gen.* **19** 2473–86
Pluhar Z, Weigert L J and Holan P 1986 *J. Phys. A: Math. Gen.* **19** 29–34
Raynal J and Conte R 1985 *J. Math. Phys.* **26** 2413–40
—— 1985 *J. Math. Phys.* **26** 2441–56
Reid M F and Butler P H 1980 *J. Phys. A: Math. Gen.* **13** 2889–901
—— 1982 *J. Phys. A: Math. Gen.* **15** 2327–35
Searle B G and Butler P H 1988 *J. Phys. A: Math. Gen.* **21** 1977–81
Zeng G J 1987 *J. Phys. A: Math. Gen.* **20** 1961–71

J. Phys. A: Math. Gen. 21 (1988) 3313–3319. Printed in the UK

# 6-$j$ for mixed symmetry triads in $K_{20}$

Barry G Searle and Philip H Butler
Physics Department, University of Canterbury, Christchurch, New Zealand

Received 26 April 1988

**Abstract.** We discuss the possible choices of conjugation and permutation matrices for mixed symmetry triads. This requires a short review of the previous choices for all triads. The primitive 6-$j$ of the finite group $K_{20}$ are then calculated as an example of a group that contains mixed symmetry triads.

## 1. Introduction

Although numerous 6-$j$ have been calculated for various groups, relatively few workers have attempted to calculate 6-$j$ with multiplicity and very few (see Zhang and Xiangzhu 1987, Gao and Chen 1985) have attempted to calculate 6-$j$ involving mixed symmetry couplings. Since mixed symmetry triads are a common occurrence in most Lie groups (exceptions are $SO_2, SO_3$ and $SU_3$) and in the symmetric groups $S_n$, for $n \geqslant 6$, these cases must eventually be analysed.

In our development of a PASCAL program to perform the calculation of 6-$j$ for a general compact group, it was necessary to design it to handle mixed symmetry triads. The program only has the selection rules for the group as information and also assumes a particular choice of permutation and conjugation matrices for the various triads. Therefore we need to study the available choices for these matrices, how the various choices are interrelated and which choice is the most convenient form from the viewpoint of both group theory and programming.

We study the various phase choices and decide on an appropriate choice for mixed symmetry triads and then check our results by calculating primitive 6-$j$ for a small finite group with mixed symmetry triads. Bickerstaff suggested that we try the K-metacyclic group of order 20 ($K_{20}$) as a test for our ideas since it was a group he had attempted but not completed due to the effort required to solve the large number of 6-$j$ containing mixed symmetry. All but one of the irreps in $K_{20}$ are one dimensional, but the non-trivial irrep has a mixed symmetry triad. The rather special nature of $K_{20}$ also gives us an opportunity to produce an example of the calculation of core 6-$j$, as discussed in a previous paper (Searle and Butler 1988).

In § 2 we look at the general case and review what is known about the various matrices and the choices that have previously been used. We discuss the possible choices for the symmetry relations of mixed symmetry triads and consider their various merits in § 3. Finally in § 4 we introduce the group $K_{20}$ and present our results for its primitive 6-$j$. These are the first 6-$j$ we have found that are strictly complex, although such cases are not unknown for the 3-$jm$ symbols (such as T to $D_2$; see Butler (1981)).

## 2. Review

Derome and Sharp (1965) introduced unitary $A, M$ and $U$ matrices to describe the symmetries of a generalised 3-$jm$ or 6-$j$ symbol for any compact group. We will review these and other results for the choice of these matrices (see Butler 1975, Bickerstaff 1981) in this section. These transformation matrices are defined with respect to their effect on the group triad of a 3-$jm$ (and hence four are required for a 6-$j$ since it can be written as a product of four 3-$jm$).

We describe the $U$ matrix first. This matrix relates 3-$jm$ symbols with alternative coupling multiplicity choices via a unitary transformation

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r}_{\mathrm{alt}} = \sum_{r'} U(\lambda_1\lambda_2\lambda_3)^{r}_{r'} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r'}. \tag{2.1}$$

The $U$ matrix therefore describes the freedom of choice we have in the value of the 3-$jm$ or 6-$j$ due to the coupling process (see Searle and Butler 1988), as distinct from the freedom in 3-$jm$ symbols due to the freedom in branching multiplicity. The freedom described by the $U$ matrix can be used to study the possible choices for the two matrices, $A$ and $M$.

The $M$ matrix gives the property of a 3-$jm$ under a column permutation (a reordering of the coupling) in the following manner:

$$\begin{pmatrix} \lambda_a & \lambda_b & \lambda_c \\ i_a & i_b & i_c \end{pmatrix}^{r} = \sum_{r'} M(\pi, \lambda_1\lambda_2\lambda_3)^{r}_{r'} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r'} \tag{2.2}$$

where $\pi$ is the permutation performed on the indices (namely $(abc) = \pi(123)$). The elements of $M$ are known as 3-$j$ phases. Alternative choices of the permutation matrices are related via the $U$ matrix as follows:

$$M'(\pi, \lambda_1\lambda_2\lambda_3) = U(\lambda_a\lambda_b\lambda_c)^{\dagger} M(\pi, \lambda_1\lambda_2\lambda_3) U(\lambda_1\lambda_2\lambda_3). \tag{2.3}$$

Derome (1966) has discussed the choices for the $M$ matrix allowed by (2.3) and has found the simplest values of $M$ that could be used. Whenever all three irreps of the triad $\lambda_1\lambda_2\lambda_3$ are not identical, $M$ can be chosen as a diagonal matrix with diagonal entries of $\pm 1$ by choosing $U(\lambda_a\lambda_b\lambda_c)$ in relation to $U(\lambda_1\lambda_2\lambda_3)$. In the case that $\lambda_1 = \lambda_2 = \lambda_3$ the $M$ matrix can be chosen block diagonal with respect to symmetry type. However, three symmetry types may occur: these are the symmetric, mixed and antisymmetric types whose occurrence is related, respectively, to the occurrence of the scalar in the symmetric, mixed and antisymmetric parts of the cube of $\lambda_1$. For the symmetric and antisymmetric parts of the cube of $\lambda_1$ ($\lambda_1 \otimes \{3\}$ and $\lambda_1 \otimes \{1^3\}$, respectively) the blocks may be chosen as $I_{[3]}$ or $-I_{[1^3]}$. The dimension of the unit matrix is the same as the multiplicity of the scalar in the appropriate part of the cube. An occurrence of the scalar in $\lambda \otimes \{21\}$ means we have an occurrence of a so-called mixed symmetry triad, where the column permutation symmetry must be represented by two-dimensional matrices of the irrep [21] of $S_3$. Our $M$ matrix for $\lambda_1 = \lambda_2 = \lambda_3\ (=\lambda)$ is then of the form

$$M(\pi, \lambda\lambda\lambda) = \begin{pmatrix} I_{[3]} & & \\ & M_{[21]} & \\ & & -I_{[1^3]} \end{pmatrix}$$

where the $M_{[21]}$ block is itself composed of the two-dimensional $S_3$ irrep matrix (for the permutation $\pi$) as blocks on its diagonal. $M_{[21]}$ has dimension of twice the multiplicity of the scalar in $\lambda \otimes \{21\}$.

The final unitary matrix to consider is the conjugation or $A$ matrix, where

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ i_1 & i_2 & i_3 \end{pmatrix}^{r*} = \sum_{r'i_1'i_2'i_3'} A(\lambda_1\lambda_2\lambda_3)_{rr'}(\lambda_1)^{i_1i_1'}(\lambda_2)^{i_2i_2'}(\lambda_3)^{i_3i_3'} \begin{pmatrix} \lambda_1^* & \lambda_2^* & \lambda_3^* \\ i_1' & i_2' & i_3' \end{pmatrix}^{r'}. \tag{2.4}$$

It has been usual to choose the $A$ matrix equal to $I$ for all couplings, although sometimes this choice does not give real coupling symbols (Sullivan 1983). The various forms of the $A$ matrix are related to the other matrices via

$$A'(\lambda_1^*\lambda_2^*\lambda_3^*) = U(\lambda_1\lambda_2\lambda_3)^{\mathrm{T}} A(\lambda_1^*\lambda_2^*\lambda_3^*) U(\lambda_1^*\lambda_2^*\lambda_3^*) \tag{2.5}$$

and

$$A'(\lambda_a\lambda_b\lambda_c) = M(\pi^{-1}\lambda_a\lambda_b\lambda_c)^{\mathrm{T}} A(\lambda_1\lambda_2\lambda_3) M(\pi\lambda_1\lambda_2\lambda_3)^{\dagger}. \tag{2.6}$$

We shall discuss further the choices of $A$ for mixed symmetry triads in the next section.

## 3. Choices of the $S_3$ irrep matrices

The usual matrix form for the two-dimensional representation of $S_3$ consists of real orthogonal matrices with the generator matrices $(12) = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$, $(123) = \frac{1}{2}\begin{pmatrix} -1 & \sqrt{3} \\ -\sqrt{3} & -1 \end{pmatrix}$. As a result, for a mixed symmetry triad pair $\lambda\lambda\lambda 1$ and $\lambda\lambda\lambda 2$, the 3-*jm* with permuted columns is a linear combination of 3-*jm* with unpermuted columns. In particular we have

$$\begin{pmatrix} \lambda & \lambda & \lambda \\ k & i & j \end{pmatrix}^1 = -\frac{1}{2}\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^1 + \frac{\sqrt{3}}{2}\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^2. \tag{3.1}$$

An alternative representation of this irrep has the second of these generators (the 3-cycle) diagonalised, giving $(12) = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$ and $(123) = \begin{pmatrix} \omega^2 & \\ & \omega \end{pmatrix}$ with $\omega = \exp(2\pi \mathrm{i}/3)$. Use of this matrix irrep would imply the use of complex 3-*jm* symbols, but the 3-*jm* with permuted columns is simply related to an unpermuted symbol, e.g.

$$\begin{pmatrix} \lambda & \lambda & \lambda \\ i & j & k \end{pmatrix}^1 = \begin{pmatrix} \lambda & \lambda & \lambda \\ j & i & k \end{pmatrix}^2 = \omega^2 \begin{pmatrix} \lambda & \lambda & \lambda \\ j & k & i \end{pmatrix}^1. \tag{3.2}$$

We wish to know whether there is any restriction on the use of either of these two choices and what effect they have on the choice of the $A$ matrix.

At this stage we impose the requirement that the product of two orthogonal or two symplectic irreps contains only orthogonal irreps. This is a restriction that is satisfied by all triads of all the classical Lie groups, point groups, symmetric groups and many finite groups. With this restriction it has been shown (Butler 1975) that the $A$ matrix is a symmetric unitary matrix which is block diagonal on symmetry type. We would like to be able to choose the $A$ matrix to be the identity, as this is consistent with most previous workers' choices.

For the choices of the $M$ matrix in § 2 it is known that we may choose $A = I$. If the real choice of the $M$ matrix for a mixed symmetry triad is used then (2.6) shows that the $A$ matrix must be a multiple of $I$. However, when we apply the generators of the complex choice of the $M$ matrix to $A$ we find

$$A(\lambda\lambda\lambda)=\begin{pmatrix}\omega & \\ & \omega^2\end{pmatrix} A(\lambda\lambda\lambda)\begin{pmatrix}\omega & \\ & \omega^2\end{pmatrix} \tag{3.3}$$

which requires that the diagonal elements of $A$ are zero. The other generator then fixes $A$ as a multiple of $\left(\begin{smallmatrix} & 1\\ 1 & \end{smallmatrix}\right)$. This skew-diagonal matrix relates one conjugated symbol to the symbol for the other multiplicity of the pair, e.g.

$$\begin{pmatrix}\lambda & \lambda & \lambda\\ j & i & k\end{pmatrix}^{1*}=\sum_{i',j',k'}(\lambda)^{ii'}(\lambda)^{jj'}(\lambda)^{kk'}\begin{pmatrix}\lambda^* & \lambda^* & \lambda^*\\ i' & j' & k'\end{pmatrix}^{2}. \tag{3.4}$$

We will choose $A=I$ and will choose the real set of permutation matrices. This will prevent the permuted forms of a symbol being complex if the symbol can be chosen real and allows the $A$ matrix to be omitted in most applications of the Racah–Wigner algebra.

## 4. Primitive 6-*j* for $K_{20}$

To test the calculation of mixed symmetry triads using the above results we looked for a small finite group that contained such a triad. The K metacyclic group of order 20 (see Biedenharn *et al* 1968, Bovier *et al* 1981) is the smallest such group. It has only one irrep that is not one dimensional and all irreps are quasi-orthogonal. All the primitive 6-*j* that occur are core (as defined by Searle and Butler 1988) and the irrep with the mixed symmetry triad is also the primitive irrep (see tables 1 and 2) where the triads 4441 and 4442 form the only mixed symmetry pair.

The one basis 6-*j* that does not contain 444$r$ is $\left\{\begin{smallmatrix}2 & 2 & 1\\ 4 & 4 & 4\end{smallmatrix}\right\}$ and is trivially solved by using the normality relation. Those few non-basis 6-*j* that do not contain any of the triads 444$r$ (see table 2) can then be readily solved by the Racah backcoupling equation.

**Table 1.** Character table.

| | Class | | | | |
|---|---|---|---|---|---|
| Irrep | E | T | S | $T^2$ | $T^3$ |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | −1 | 1 | 1 | −1 |
| 2 | 1 | i | 1 | −1 | −i |
| 3 | 1 | −i | 1 | −1 | i |
| 4 | 4 | 0 | −1 | 0 | 0 |

**Table 2.** Table of 3-*j*.

| | | |
|---|---|---|
| 0 0 0 0+ | 4 4 0 0+ | 4 4 4 0+ |
| 4 4 4 1+ | 4 4 4 2− | 1 1 0 0+ |
| 1 4 4 0+ | 2 4 4 0− | 2 2 1 0+ |
| | 3 2 0 0+ | |

Solving for the large number of remaining 6-$j$ is more complicated since the column permutation of any 6-$j$ is related to a linear combination of some others (even when a permutation does not seem to alter the irreps, it will still permute the multiplicity indices).

For example, a (23) interchange of $\left\{\begin{smallmatrix}4&4&4\\4&4&4\end{smallmatrix}\right\}_{0011}$ is $\left\{\begin{smallmatrix}4&4&4\\4&4&4\end{smallmatrix}\right\}_{0101}$ which, by the symmetry relations, is equal to

$$\sum_{ab} M(23,444)_{aa'} M(23,444)_{bb'} \begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{00a'b'}$$

since $M(23,444)_{0a}=\delta_{0a}$.

By making use of the results of Newmarch (1983) to find out which 6-$j$ should be considered independent, of which there were 14, we were able to solve the independent set. We chose

$$\begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{0000} \begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{0011} \begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{0012} \begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{0111}$$

$$\begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{0112} \begin{Bmatrix}4&4&4\\4&4&4\end{Bmatrix}_{1111} \begin{Bmatrix}1&4&4\\4&4&4\end{Bmatrix}_{0000} \begin{Bmatrix}1&4&4\\4&4&4\end{Bmatrix}_{0010}$$

$$\begin{Bmatrix}1&4&4\\4&4&4\end{Bmatrix}_{0110} \begin{Bmatrix}1&4&4\\4&4&4\end{Bmatrix}_{0120} \begin{Bmatrix}2&4&4\\4&4&4\end{Bmatrix}_{0000} \begin{Bmatrix}2&4&4\\4&4&4\end{Bmatrix}_{0010}$$

$$\begin{Bmatrix}2&4&4\\4&4&4\end{Bmatrix}_{0110} \begin{Bmatrix}2&4&4\\4&4&4\end{Bmatrix}_{0120}$$

as our independent set. By using both the orthogonality and Racah backcoupling equations (with normality for the few basis 6-$j$) we were able to obtain sufficient independent equations to resolve the remaining 6-$j$. The Racah backcoupling equations gave the necessary extra information because they include information on the symmetry of the triads. The simultaneous equations for the independent 6-$j$ were created and solved by the algebraic program REDUCE, as were the symmetry relations for the related 6-$j$. The results were put into the Biedenharn–Elliott equation as an independent check that we had correctly solved the various equations. The set of 6-$j$ are given in table 3, in the same format as used for the tables of Butler (1981).

The 6-$j$ for this group, $K_{20}$, are the first set of 6-$j$ to be calculated where the 6-$j$ are strictly complex. It is impossible to find a different $U$ matrix that will give pure real or imaginary values without producing imaginary values for some of the real 6-$j$. A change of multiplicity separation for the mixed symmetry pair affects 3-$jm$ as in (2.1) and will affect 6-$j$ similarly, except that there are four matrices, one for the multiplicity of each triad (see equation (4.1) of Searle and Butler (1988)). Any such change merely moves the pure real, pure imaginary and strictly complex values around table 3.

## 5. Conclusion

This paper has reviewed some of the knowledge about the symmetry matrices for the generalised 3-$jm$ and 6-$j$ symbols. We have then discussed the possible choices for the complex conjugation matrix given various choices of permutation matrix for a mixed symmetry triad. These choices have then been used in calculating the primitive 6-$j$ of the finite group, $K_{20}$.

**Table 3.** Table of 6-*j*.

<u>0 0 0</u>
0 0 0 0000 $+1$

<u>4 4 0</u>
0 0 4 0000 $+1/2$
4 4 0 0000 $+1/4$

<u>4 4 4</u>
4 4 0 0000 $+1/4$
4 4 0 0001 0
4 4 0 0002 0
4 4 0 0011 $+1/4$
4 4 0 0012 0
4 4 0 0022 $-1/4$
4 4 4 0000 $+1/6$
4 4 4 0001 0
4 4 4 0002 0
4 4 4 0010 0
4 4 4 0011 $-1/12$
4 4 4 0012 0
4 4 4 0020 0
4 4 4 0021 0
4 4 4 0022 $+1/12$
4 4 4 0100 0
4 4 4 0101 $+1/24$
4 4 4 0102 $+1/8\sqrt{3}$
4 4 4 0110 $+1/24$
4 4 4 0111 $+1/12\sqrt{2}$
4 4 4 0112 $-1/4\sqrt{6}$
4 4 4 0120 $-1/8\sqrt{3}$
4 4 4 0121 $+1/4\sqrt{6}$
4 4 4 0122 $+1/12\sqrt{2}$
4 4 4 0200 0
4 4 4 0201 $+1/8\sqrt{3}$
4 4 4 0202 $-1/24$
4 4 4 0210 $-1/8\sqrt{3}$
4 4 4 0211 0
4 4 4 0212 $-1/6\sqrt{2}$
4 4 4 0220 $-1/24$
4 4 4 0221 $-1/6\sqrt{2}$
4 4 4 0222 0
4 4 4 1000 0
4 4 4 1001 $+1/24$
4 4 4 1002 $-1/8\sqrt{3}$
4 4 4 1010 $+1/24$
4 4 4 1011 $+1/12\sqrt{2}$
4 4 4 1012 $+1/4\sqrt{6}$
4 4 4 1020 $+1/8\sqrt{3}$
4 4 4 1021 $-1/4\sqrt{6}$
4 4 4 1022 $+1/12\sqrt{2}$
4 4 4 1100 $-1/12$
4 4 4 1101 $+1/12\sqrt{2}$
4 4 4 1102 0
4 4 4 1110 $+1/12\sqrt{2}$
4 4 4 1111 $+1/8$
4 4 4 1112 0
4 4 4 1120 0
4 4 4 1121 0

<u>4 4 4</u> (continued)
4 4 4 1122 $+1/24$
4 4 4 1200 0
4 4 4 1201 $-1/4\sqrt{6}$
4 4 4 1202 $-1/6\sqrt{2}$
4 4 4 1210 $+1/4\sqrt{6}$
4 4 4 1211 0
4 4 4 1212 $+1/24$
4 4 4 1220 $-1/6\sqrt{2}$
4 4 4 1221 $+1/24$
4 4 4 1222 0
4 4 4 2000 0
4 4 4 2001 $-1/8\sqrt{3}$
4 4 4 2002 $-1/24$
4 4 4 2010 $+1/8\sqrt{3}$
4 4 4 2011 0
4 4 4 2012 $-1/6\sqrt{2}$
4 4 4 2020 $-1/24$
4 4 4 2021 $-1/6\sqrt{2}$
4 4 4 2022 0
4 4 4 2100 0
4 4 4 2101 $+1/4\sqrt{6}$
4 4 4 2102 $-1/6\sqrt{2}$
4 4 4 2110 $-1/4\sqrt{6}$
4 4 4 2111 0
4 4 4 2112 $+1/24$
4 4 4 2120 $-1/6\sqrt{2}$
4 4 4 2121 $+1/24$
4 4 4 2122 0
4 4 4 2200 $+1/12$
4 4 4 2201 $+1/12\sqrt{2}$
4 4 4 2202 0
4 4 4 2210 $+1/12\sqrt{2}$
4 4 4 2211 $+1/24$
4 4 4 2212 0
4 4 4 2220 0
4 4 4 2221 0
4 4 4 2222 $+1/8$
1 4 4 0000 $-1/12$
1 4 4 0001 $+1/6\sqrt{2}$
1 4 4 0002 $+1/2\sqrt{6}$
1 4 4 1000 $+1/6\sqrt{2}$
1 4 4 1001 $+5/24$
1 4 4 1002 $-1/8\sqrt{3}$
1 4 4 2000 $+1/2\sqrt{6}$
1 4 4 2001 $-1/8\sqrt{3}$
1 4 4 2002 $+1/8$
4 1 4 0000 $-1/12$
4 1 4 0001 $+1/6\sqrt{2}$
4 1 4 0002 $-1/2\sqrt{6}$
4 1 4 0100 $+1/6\sqrt{2}$
4 1 4 0101 $+5/24$
4 1 4 0102 $+1/8\sqrt{3}$
4 1 4 0200 $-1/2\sqrt{6}$
4 1 4 0201 $+1/8\sqrt{3}$
4 1 4 0202 $+1/8$
4 4 1 0000 $-1/12$
4 4 1 0001 $-1/3\sqrt{2}$

<u>4 4 4</u> (continued)
4 4 1 0002 0
4 4 1 0010 $-1/3\sqrt{2}$
4 4 1 0011 $+1/12$
4 4 1 0012 0
4 4 1 0020 0
4 4 1 0021 0
4 4 1 0022 $+1/4$
2 4 4 0000 $+1/12$
2 4 4 0001 $(1+3i)/12\sqrt{2}$
2 4 4 0002 $(1-i)/4\sqrt{6}$
2 4 4 1000 $(1-3i)/12\sqrt{2}$
2 4 4 1001 $+1/24$
2 4 4 1002 $(1+2i)/8\sqrt{3}$
2 4 4 2000 $(1+i)/4\sqrt{6}$
2 4 4 2001 $(1-2i)/8\sqrt{3}$
2 4 4 2002 $+1/8$
4 2 4 0000 $+1/12$
4 2 4 0001 $(1-3i)/12\sqrt{2}$
4 2 4 0002 $-(1+i)/4\sqrt{6}$
4 2 4 0100 $(1+3i)/12\sqrt{2}$
4 2 4 0101 $+1/24$
4 2 4 0102 $(-1+2i)/8\sqrt{3}$
4 2 4 0200 $(-1+i)/4\sqrt{6}$
4 2 4 0201 $-(1+2i)/8\sqrt{3}$
4 2 4 0202 $+1/8$
4 4 2 0000 $+1/12$
4 4 2 0001 $-1/6\sqrt{2}$
4 4 2 0002 $+i/2\sqrt{6}$
4 4 2 0010 $-1/6\sqrt{2}$
4 4 2 0011 $+1/6$
4 4 2 0012 $+i/4\sqrt{3}$
4 4 2 0020 $-i/2\sqrt{6}$
4 4 2 0021 $-i/4\sqrt{3}$
4 4 2 0022 0

<u>4 1 4</u>
4 4 4 0000 $-1/12$
4 4 4 0010 $+1/6\sqrt{2}$
4 4 4 0020 $-1/2\sqrt{6}$
4 4 4 1000 $+1/6\sqrt{2}$
4 4 4 1010 $+5/24$
4 4 4 1020 $+1/8\sqrt{3}$
4 4 4 2000 $-1/2\sqrt{6}$
4 4 4 2010 $+1/8\sqrt{3}$
4 4 4 2020 $+1/8$

<u>4 2 4</u>
4 4 4 0000 $+1/12$
4 4 4 0010 $(1+3i)/12\sqrt{2}$
4 4 4 0020 $(-1+i)/4\sqrt{6}$
4 4 4 1000 $(1-3i)/12\sqrt{2}$
4 4 4 1010 $+1/24$
4 4 4 1020 $-(1+2i)/8\sqrt{3}$
4 4 4 2000 $-(1+i)/4\sqrt{6}$
4 4 4 2010 $(-1+2i)/8\sqrt{3}$
4 4 4 2020 $+1/8$

**Table 3.** (continued)

| | | |
|---|---|---|
| 4 4 1 | 1 4 4 | 2 4 4 (continued) |
| 4 4 4 0000−1/12 | 0 4 4 0000+1/4 | 4 4 4 0010 (1+3i)/12√2 |
| 4 4 4 0100−1/3√2 | 4 1 0 0000+1/2 | 4 4 4 0020 (1−i)/4√6 |
| 4 4 4 0200 0 | 4 4 4 0000−1/12 | 4 4 4 0100 (1+3i)/12√2 |
| 4 4 4 1000−1/3√2 | 4 4 4 0010+1/6√2 | 4 4 4 0110+1/24 |
| 4 4 4 1100+1/12 | 4 4 4 0020+1/2√6 | 4 4 4 0120 (1+2i)/8√3 |
| 4 4 4 1200 0 | 4 4 4 0100+1/6√2 | 4 4 4 0200 (1+i)/4√6 |
| 4 4 4 2000 0 | 4 4 4 0110+5/24 | 4 4 4 0210 (1−2i)/8√3 |
| 4 4 4 2100 0 | 4 4 4 0120−1/8√3 | 4 4 4 0220+1/8 |
| 4 4 4 2200+1/4 | 4 4 4 0200+1/2√6 | 1 4 4 0000−1/4 |
| | 4 4 4 0210−1/8√3 | 2 4 4 0000+1/4 |
| | 4 4 4 0220+1/8 | 3 4 4 0000+1/4 |
| 4 4 2 | 1 4 4 0000+1/4 | |
| 4 4 4 0000+1/12 | | 2 2 1 |
| 4 4 4 0100−1/6√2 | 1 1 0 | 0 1 2 0000+1 |
| 4 4 4 0200+i/2√6 | 0 0 1 0000+1 | 4 4 4 0000+1/2 |
| 4 4 4 1000−1/6√2 | 1 1 0 0000+1 | 2 3 1 0000+1 |
| 4 4 4 1100+1/6 | | 3 2 0 0000+1 |
| 4 4 4 1200+i/4√3 | 2 4 4 | |
| 4 4 4 2000−i/2√6 | 0 4 4 0000−1/4 | 3 2 0 |
| 4 4 4 2100−i/4√3 | 4 2 0 0000−1/2 | 0 0 2 0000+1 |
| 4 4 4 2200 0 | 4 4 4 0000+1/12 | 3 3 0 0000+1 |

## References


Bickerstaff R P 1981 *PhD thesis* University of Canterbury, New Zealand

Biedenharn L C, Brouwer W and Sharp W T 1968 *The Algebra of Representations of some Finite Groups* (Houston: Rice University)

Bovier A, Lüling M and Wyler D 1981 *J. Math. Phys.* **22** 1536–42

Butler P H 1975 *Phil. Trans. R. Soc.* A **277** 545–85

—— 1981 *Point Group Symmetry Applications: Methods and Tables* (New York: Plenum)

Derome J R 1986 *J. Math. Phys.* **7** 612–5

Derome J R and Sharp W T 1965 *J. Math. Phys.* **6** 1584–90

Gao M-J and Chen J-Q 1985 *J. Phys. A: Math. Gen.* **18** 189–213

Newmarch J D 1983 *J. Math. Phys.* **24** 451–6

Searle B G and Butler P H 1988 *J. Phys A: Math. Gen.* **21** 3041–50

Sullivan J J 1983 *J. Math. Phys.* **24** 424–40

Zhang Q and Xiangzhu L 1987 *J. Phys. A: Math. Gen.* **20** 6185–96